BNAIC 2008

Proceedings of BNAIC 2008, the twentieth Belgian-Dutch Artificial Intelligence Conference

Enschede/Bad Boekelo, October 30-31, 2008

Anton Nijholt, Maja Pantic, Mannes Poel and Hendri Hondorp (eds.)

BNAIC 2008

Belgian-Dutch Conference on Artificial Intelligence

PROCEEDINGS OF THE TWENTIETH
BELGIAN-DUTCH CONFERENCE ON ARTIFICIAL INTELLIGENCE

Enschede, October 30-31, 2008

Anton Nijholt, Maja Pantic, Mannes Poel and Hendri Hondorp (eds.)

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Nijholt, A., Pantic, M., Poel, M., Hondorp, G.H.W.

Belgian-Dutch Conference on Artificial Intelligence (BNAIC) 2008
Proceedings of the twentieth Belgian-Dutch Conference on Artificial Intelligence
A. Nijholt, M. Pantic, M. Poel, G.H.W. Hondorp (eds.)
Enschede, Universiteit Twente, Faculteit Elektrotechniek, Wiskunde en Informatica

ISSN 1568–7805

trefwoorden: AI for Ambient Intelligence, AI for Games & Entertainment, Embodied Artificial Intelligence, Intelligent Agents & Multi-agent Systems, Knowledge Representation, Knowledge Management & Knowledge-based Systems, Knowledge Discovery and Data Mining, Logic and Logic Programming, Machine Learning, Evolutionary Algorithms & Neural Networks, Natural Language and Speech Processing, AI Applications.

© Copyright 2008; Universiteit Twente, Enschede

Book orders:
Ms. C. Bijron
University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
P.O. Box 217
NL 7500 AE Enschede
tel: +31 53 4893740
fax: +31 53 4893503
Email: bijron@cs.utwente.nl

Omslag: Zouthuisje en -boortoren in Boekelo.
Fotografie: Alice Vissers-Schotmeijer
Ontwerp: Ronald Poppe

Druk- en bindwerk: Xerox Global Services

Preface

This book contains the proceedings of the 20th edition of the Belgian-Netherlands Conference on Artificial Intelligence. The conference was organized by the Human Media Interaction group of the University of Twente. As usual, the conference was under the auspices of the Belgian-Dutch Association for Artificial Intelligence (BNVKI) and the Dutch Research School for Information and Knowledge Systems (SIKS). The conference aims at presenting an overview of state-of-the-art research in artificial intelligence in Belgium and the Netherlands, but does not exclude contributions from other countries. The received submissions show that AI researchers in Belgium and the Netherlands continue to work actively in many different areas of artificial intelligence and are open for new developments in technology and society.

The annual BNAIC conference is the main meeting place for artificial intelligence researchers and practioners in Belgium and the Netherlands. Therefore we did not change the tradition that beside the sessions with accepted regular papers describing original work, there are also short papers describing work published elsewhere and papers describing demonstrations. We received 108 submissions consisting of 44 regular full papers, 53 short papers, and 11 system demonstrations. We are grateful to the programme committee members who carefully reviewed all submissions. A small committee chaired by the conference chairs made the final decisions. The acceptance rate of the regular papers was 80%. Of the short papers 75% was presented in oral sessions, the others were presented in poster sessions.

As mentioned, this is the twentieth BNAIC. This is not completely true. The series started as the Dutch Artificial Intelligence Conferences (NAIC: Nederlandse Artificiële Intelligentie Conferentie) and in 1999 the first BNAIC was organized. So, we can decide to celebrate the twentieth (B)NAIC or the tenth BNAIC this year. Previous (B)NAICS were organized in Amsterdam (1988), Enschede (1989), Kerkrade (1990), Amsterdam (1991), Delft (1992), Enschede (1993), Rotterdam (1995), Utrecht (1996), Antwerpen (1997), Amsterdam (1998), Maastricht (1999), Kaatsheuvel (2000), Amsterdam (2001), Leuven (2002), Nijmegen (2003), Groningen (2004), Brussels (2005), Namur (2006), and Utrecht (2007).

Obviously, a twentieth edition asks for a special location. We found it at Resort Bad Boekelo, a hotel and conference centre near Enschede with beautiful facilities and in beautiful surroundings, giving participants the opportunity to merge scientific and recreational activities such as walking in the woods, diving in the (indoor) swimming pool and visiting the sauna. Most of the participants stayed at least one night in this resort, making it possible to have lively discussions accompanied by, among other things, live music and local beer specialities.

The conference was sponsored by Koninklijke Nederlandse Akademie van Wetenschappen (KNAW), Vereniging Werkgemeenschap Informatiewetenschap, Delft Cooperation on Intelligent Systems (D-CIS), Dutch Research School for Information and Knowledge Systems (SIKS), Netherlands Organisation for Scientific Research (NWO), Stichting Knowledge-Based Systems (SKBS), SKF Benelux, Belgium-Netherlands Association for Artificial Intelligence, Centre of Telematics and Information Technology (CTIT), and the Human Media Interaction (HMI) research group of the University of Twente.

There were many people involved in the organization of this conference and we cannot mention them all. We gratefully acknowledge help from BNVKI board members and previous organizers. Mannes Poel took responsibility for the review process; Hendri Hondorp took care of the website and, as usual, did a perfect job compiling the proceedings. Cover design by Ronald Poppe and Alice Vissers. Paul van der Vet for awards, general advice and, together with Lynn Packwood and Theo Huibers, sponsor acquisition. Lynn also took care of financial administration. Social events were the responsibility of Betsy van Dijk and Wim Fikkert, posters and demonstrations were organized by Thijs Verschoor and Ronald Poppe. Administration, registration and overall organisation were done by Charlotte Bijron and Alice Vissers.

Finally, we thank our invited speakers, Wolfgang Wahlster (DFKI, Saarbrücken, Germany) with a talk on “Anthropomorphic Interfaces for the Internet of Things” and Ruth Aylett (Heriot-Watt University, Edinburgh, UK) with a talk on “Planning stories - emergent narrative or universal plans?”

Anton Nijholt, Maja Pantic — Enschede, September 2008

Previous BNAIC Conferences

1988	Amsterdam	1999	Maastricht
1989	Twente	2000	Kaatsheuvel
1990	Kerkrade	2001	Amsterdam
1991	Amsterdam	2002	Leuven
1992	Delft	2003	Nijmegen
1993	Twente	2004	Groningen
1995	Utrecht	2005	Brussels
1996	Utrecht	2006	Namur
1997	Antwerpen	2007	Utrecht
1998	Amsterdam		

Committees BNAIC 2008

General and Program Chairs

Anton Nijholt (University of Twente) Maja Pantic (Imperial College London, University of Twente)

Programme Committee

Ameen Abu-Hanna	Tony Belpaeme	Cor Bioch	Hendrik Blockeel
Antal Van den Bosch	Tibor Bosse	Frances Brazier	Jan Broersen
Maurice Bruynooghe	Walter Daelemans	Marc Denecker	Frank Dignum
Virginia Dignum	Marco Dorigo	Kurt Driessens	Bob Duin
Ulle Endriss	Pascal Gribomont	Frank van Harmelen	Jaap van den Herik
Tom Heskes	Koen Hindriks	Zhisheng Huang	Jean-Marie Jacquet
Catholijn Jonker	Walter Kosters	Wojtek Kowalczyk	Ben Krose
Bart Kuijpers	Peter Lucas	Elena Marchiori	John-Jules Meyer
Ann Nowe	Eric Postma	Rob Potharst	Han La Poutre
Peter van der Putten	Jan Ramon	Birna van Riemsdijk	Maarten de Rijke
Leon Rothkrantz	Pierre-Yves Schobbens	Martijn Schut	Khalil Sima'an
Maarten van Someren	Ida Sprinkhuizen-Kuijper	Yao-Hua Tan	Dirk Thierens
Leon van der Torre	Karl Tuyls	Katja Verbeeck	Frans Voorbraak
Louis Vuurpijl	Mathijs de Weerdt	Ton Weijters	Michiel van Wezel
Wim Wiegerinck	Marco Wiering	Jef Wijsen	Cees Witteveen
Jelle Zuidema			

Organization Committee

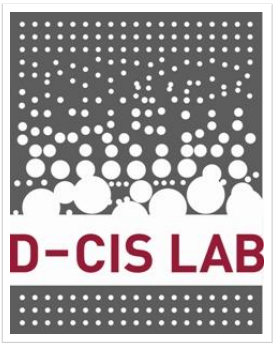
Alice Vissers	Betsy van Dijk	Charlotte Bijron	Dirk Heylen
Hendri Hondorp	Lynn Packwood	Mannes Poel	Mariët Theune
Paul van der Vet	Ronald Poppe	Thijs Verschoor	Wim Fikkert

Sponsors

[1]

[2]

[3]



[4]

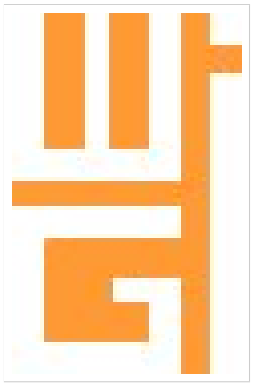
[5]

[1] http://www.ctit.utwente.nl
[2] http://www.cs.unimaas.nl/~bnvki
[3] http://www.siks.nl
[4] http://www.d-cis.nl
[5] http://hmi.ewi.utwente.nl

Sponsors (continued)

[1]

[2]



[3]

[4]

[1] http://www.skf.com
[2] http://www.nwo.nl
[3] http://www.informatiewetenschap.org
[4] http://www.knaw.nl

Contents

Full Papers

Extended Abstracts

Demonstrations

Full Papers

BNAIC 2008

Actor-Agent Based Approach to Train Driver Rescheduling

Erwin J.W. Abbink[a] David G.A. Mobach[b] Pieter J. Fioole[a] Leo G. Kroon[a,c]
Niek J.E. Wijngaards[b] Eddy H.T. van der Heijden[b]

[a] *Netherlands Railways, NSR Logistics Innovation, P.O. Box 2025, 3500 HA Utrecht*
[b] *D-CIS Lab / Thales Research & Technology NL, P.O. Box 90, 2600 AB Delft*
[c] *Rotterdam School of Management, Erasmus University Rotterdam P.O. Box 1738, NL-3000 DR Rotterdam*

Abstract

This paper describes the design of a socio-technical research system for the purpose of rescheduling train drivers in the event of disruptions. The research system is structured according to the Actor-Agent paradigm: Here agents assist in rescheduling tasks of train drivers. Coordination between agents is based on a team formation process in which possible rescheduling alternatives can be evaluated, based on constraints and preferences of involved human train drivers and dispatchers. The research aim is to explore the effectiveness of a decentralized, flexible actor-agent based approach to crew rescheduling. The research system is realized using the Cougaar framework and includes actual rolling stock schedule data and driver duty data. The current reduced-scale version shows promising results for the full-scale version end 2008.

1 Netherlands Railways: Planning and Rescheduling

Applied research on advanced autonomous systems in a real-world domain provides a stimulating environment to demonstrate 'state of the art' research results and address the encountered pragmatic and fundamental challenges. The cooperation between Netherlands Railways (NS) and the D-CIS Lab addresses a complex situation: how to reschedule tasks of train drivers in response to disruptions in their schedules. The aim is to arrive at a research system for ongoing experimentation by NS. This paper provides a brief overview of the problem domain, the design of the actor-agent based solution, and the current midway implementation with a brief comparison with related literature.

1.1 Planning

The railway operations of Netherlands Railways (NS) are based on an extensive planning process. After the planning process, the plans are carried out in the real-time operations. Preferably, the plans are carried out exactly as scheduled. However, in real-time operations plans have to be updated permanently in order to deal with delays of trains and larger disruptions of the railway system.

In the operational planning process of NS, the timetable is planned first. The rolling stock and crew schedules are planned consecutively. The timetable consists of the line system and arrival and departure times of trains. The departure and arrival times of trains are such that the timetable is cyclic with a cycle time of one hour. The rolling stock planning supplies each train in the timetable with sufficient rolling stock for transporting the forecasted number of passengers. The crew scheduling stage supplies each train with a train driver and with sufficient conductors. In the past years, NS has successfully applied novel Operations Research models to significantly improve the crew scheduling process [1]. In this paper we focus on an actor-agent based approach for rescheduling of *train drivers*.

NS train drivers operate from 29 crew depots. Each day a driver carries out a number of tasks, which means that he/she operates a train on a trip from a certain start location and start time to a certain end location and end time. The trips of the trains are defined by the timetable. The tasks of the drivers have

been organized in a number of duties, where each duty represents the tasks to be carried out by a single driver on a single day. Each duty starts in a crew base, and a hard constraint is that the duty returns to the same crew base within a limited period of time. Also several other constraints must be satisfied by the duties, such as the presence of a meal break at an appropriate time and location, and an average working time per depot of at most 8:00 hours. Initially, duties are anonymous, which means that the allocation of drivers to duties is still to be made. The latter is handled by the crew rosters, which describe the sequence of duties that are carried out by the drivers on consecutive days.

The total number of train drivers is about 3000. Each day, about 1000 duties are carried out. Furthermore, at any moment in time, the number of active duties at that moment is about 300. Note that, apart from the operational planning process described above, there is also a strategic planning process, which falls outside the scope of this paper.

1.2 Timetable and rolling stock rescheduling

In case of delays or a disruption of the railway system in the real-time operations, the original timetable, rolling stock circulation and crew duties may become infeasible. A disruption may be due to an incident, or a breakdown of infrastructure or rolling stock. On the Dutch rail network, on average three complete blockages of a route occur per day. Rescheduling is required to cope with these situations.

For example, consider a train line between stations S1, S2, S3 and S4. Under normal circumstances, trains on this line are operated from S1 to S4 and from S4 to S1. A train that arrives in S4 returns to S1, and vice versa. However, if there is a breakdown of the infrastructure between S2 and S3, then temporarily no trains can be operated between these stations. In such a situation the timetable is modified by cancelling trips between these stations. Furthermore, the standard strategy to reschedule the rolling stock is to introduce returns of trains in stations S2 and S3: a train that arrives in S2 from S1 returns to S1, and a train that arrives in S3 from S4 returns to S4 (see Figure 1a).

Due to the cyclic nature of the timetable and the structure of the rolling stock circulation the basic principles of timetable and rolling stock rescheduling are rather straightforward. Since crew duties do not have a cyclic nature, crew rescheduling is more complicated (see 1.3). A further complicating issue in a disrupted situation is the fact that the exact duration of the disruption is usually not known exactly. That is, the initial estimate of the duration of the disruption often turns out to be incorrect. As a consequence, the rescheduling process must be carried out several times.

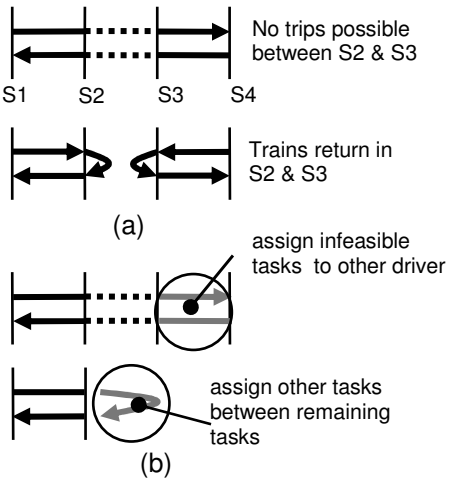


Figure 1: Disruption consequences

1.3 Train driver rescheduling

Due to delays of trains or rescheduling of the timetable and the rolling stock a number of duties of train drivers may become infeasible. An infeasibility of a duty is due to a time conflict or a location conflict. In both cases, a conflict occurs between two consecutive tasks in the duty.

A time conflict occurs if the end location of the first task coincides with the start location of the next one, but the end time of the first task is later than the start time of the second one. This is due to a delay of the train corresponding to the first task. If after the first task the duty prescribes a transfer of the driver to a task on another train, then the driver is too late for carrying out the second task. In order to make the duties more robust against such time conflicts, they contain a certain buffer time between each pair of consecutive tasks that are carried out on different trains.

A location conflict occurs if the end location of a task in a duty differs from the start location of the next task in the duty. This may be due to the fact that some tasks in the original duty were cancelled because of a disruption. Again, consider the example of the line between stations S1 to S4 (see Figure 1b). If an original duty contains the tasks S1-S2, S2-S3, S3-S4 on a train in one direction, and the tasks

S4-S3, S3-S2, and S2-S1 on a train back, but the tasks S2-S3 and S3-S2 have been cancelled, then the duty has two location conflicts: the tasks S3-S4 and S4-S3 have to be transferred to another duty, since they cannot be carried out by the originally assigned driver. Furthermore, the resulting hole in the duty between the tasks S1-S2 and S2-S1 can be filled with other tasks. To get more flexibility in the rescheduling process, the final task S2-S1 in the duty may also be transferred to another duty.

Note that at least the duties that are directly affected by the disruption must be rescheduled. But usually also a number of other duties are rescheduled in case of a disruption. Without rescheduling these additional duties, it may be impossible to find an appropriate solution satisfying the operational rules. Moreover, in several crew depots a number of stand-by train drivers are available that may take over parts of duties of other drivers in case of a disruption of the railway system. If it is still impossible to find an appropriate driver for each trip in the modified timetable, then the consequence is that the uncovered trips will have to be cancelled. This requires the rolling stock to be rescheduled again.

Currently, the rescheduling process is carried out in four operational control centres of NS: each region has its own operational control centre. However, this organization requires extensive communication between these centres, since many trains and duties operate in more than one region. In order to reduce the communication between the control centres, the process will be reorganized, and carried out in one control centre in the near future.

2 Socio-Technical Design

In this section the actor-agent based solution to train driver rescheduling is described. First, the context of the system is described; after this, the main principle underlying the actor-agent based rescheduling process is introduced. Subsequently, the actors and agents are introduced, as well as the concept of agent-teaming for rescheduling. Finally, the team formation process is described. Throughout this section, a train-network disruption scenario is used to illustrate the introduced concepts.

2.1 Design context

The system is designed according to the actor-agent paradigm [8], which explicitly recognizes both human actors and artificial agents as equivalent team members, each fulfilling their respective roles in order to reach the team objectives. The actor-agent based design process provides the system with several useful global system characteristics. First, the decentralized approach in which agents use local knowledge, world views, and interactions, contributes to an open system design. This openness facilitates easy reconfiguration and/or adaptation to changing system requirements. Second, combining humans and agents within the system design allows for integrating them at their appropriate abstraction levels: Human dispatchers at the strategic/management level, train drivers at the level of defining and guarding their personal interests, and their respective agents at the level of implementing the strategic/management decisions and resolving actual schedule conflicts.

The prototype system currently being developed focuses on rescheduling train driver duties in real-time over the course of a single day. The term *schedule* is used in the remainder of this paper to indicate duties assigned to train drivers. It is assumed that any rolling stock plan modifications to cope with disruptions (see Section 1.2) have been implemented, and a new rolling stock plan is in place, to which the driver schedules must be adapted.

2.2 Principle: Resolving conflicts by exchanging tasks

The basic principle underlying the solution process is that of *task exchange*. Each driver's schedule consists of a number of tasks (i.e. train driving activities). If in the event of a disruption a driver can no longer perform one or more tasks due to a schedule conflict (location-based or time-based), these tasks are taken over by another driver. In turn, this driver may have to hand over tasks which conflict with the newly accepted tasks to another driver.

To further illustrate the principle, a small scenario is introduced which is used throughout the remainder of this section: The scenario consists of a delayed train (+30 minutes), as a result of which a single driver (designated 'Dordrecht 109', or *Ddr-109*) is directly affected. Figure 2 shows the effect of

the disruption on the schedule of the affected driver: As a consequence of the delay the driver arrives too late in *Asd* (Amsterdam) on trip A, which results in a time-based conflict with trip B.

The solution process results in a number of drivers exchanging tasks, eventually resulting in all conflicting tasks being reassigned to other drivers. In the following sections, the actors and agents involved in the exchange process are introduced, and the exchange process is described in more detail.

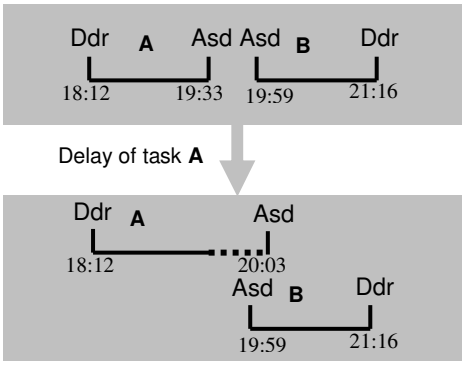


Figure 2: Schedule conflict

2.3 Actors, agents and teams

The following actors and agents involved in the rescheduling process are distinguished (see Figure 3):

- **Dispatcher-actor**: Responsible for the overall rescheduling process. When a disruption occurs, the dispatcher specifies global rescheduling parameters (e.g. number of stand-by drivers that may be used, maximum overtime allowed for driver-agents), monitors the rescheduling process, and evaluates the proposed rescheduling solutions.
- **Driver-actor**: Responsible for execution of a schedule. A driver-actor imposes constraints on the rescheduling process based on the preferences he/she may have with respect to performing his/her duties. These constraints can be hard (e.g. familiarity with rolling stock types) or soft (preferences for certain lines). Each driver-actor is associated with a driver-agent with which he/she interacts in order to reflect personal preferences in the rescheduling process.
- **Dispatcher-agent**: Presents a dispatcher-actor with a management view on the rescheduling process and coordinates the rescheduling process on the level of the team formation process. Rescheduling proposals are presented by the dispatcher-agent to the dispatcher-actor.
- **Driver-agent**: Responsible for resolving conflicts arising in schedules due to disruptions. Each driver-agent is linked to a specific driver-actor which it represents in the rescheduling process. Driver-agents engage in a team formation process in order to find a suitable team configuration in which tasks are exchanged. Driver-agents directly affected by disruptions assume the role of team leader, and other driver-agents join teams when they can help to solve a conflict.
- **Network/duty-analyzer-agent**: Maintains an up-to-date view of the rail network, reflecting any changes in timetable and rolling stock due to disruptions. Driver-agents interact with a duty-analyzer-agent to determine whether it is possible to incorporate tasks of other agents into their existing schedules (i.e. whether it is possible to take part in a task exchange). To this end, the duty-analyzer-agent attempts to find a route for the driver-agent through the rail network on the currently available timetable and rolling stock. Adding tasks to an existing schedule may entail dropping existing tasks from the schedule. The duty-analyzer agent determines the minimum number of tasks to drop, thus maintaining as much of the original schedule as possible.

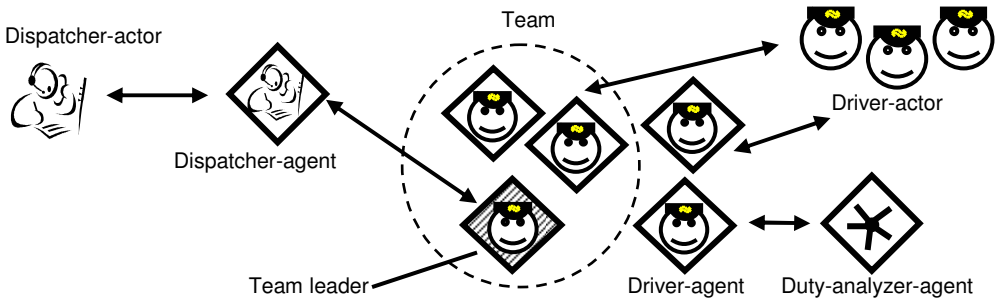


Figure 3: Overview of actors and agents

2.4 Team formation process

The coordination mechanism used by driver-agents to find proper sequences of task exchanges is based on team formation: When a disruption occurs, *all* driver-agents are informed of the impact of the disruption on the current timetable and rolling stock schedule by the dispatcher-agent. A driver-agent affected by the disruption starts a new team and invites other agents to join the team. Driver-agents will accept the invitation if a task exchange is possible. In turn, these agents may invite other agents if additional task exchanges are necessary. Ultimately, a team leader compares and chooses the best team configuration. For reasons of space, the configuration protocol is not described in detail in this paper. Instead, the protocol is described below in terms of the four main phases.

Phase 1: Discovery: When a driver-agent determines that a disruption directly affects the driver's schedule (i.e. the specified train service is associated with a task in the driver's schedule), the agent assumes the role of *team leader*. The responsibility of a team leader is to establish and analyze possible team configurations which resolve the agent's schedule conflicts. All team leaders report their new team leader status to the dispatcher-agent.

In the example scenario, driver-agent Ddr-109 has determined that the delayed task leads to a conflict in its schedule and announces itself as new team leader. The task that it needs to exchange in order to resolve the conflict consists of the trip *Asd-Ddr* (see Figure 4).

Phase 2: Team extension: In this phase, a team leader announces the conflicting tasks to other driver-agents. Each driver-agent then determines whether the announced tasks can be fitted into their schedules. This starts a *recursive team extension* process in which each team is extended with additional team members able to take over tasks from agents already participating in the team: In case a driver-agent has determined that tasks can be taken over conditionally and that it is worthwhile to join the team, the set of new conflicting tasks of this agent is again announced to other driver-agents. This leads to a recursive addition of layers of team members to the team, resulting in a team consisting of multiple task exchange configurations. In this team extension process, it is possible for driver-agents to participate multiple times in task exchanges within the same team (and in other teams). This allows for teams to discover configurations in which driver-agents 'trade' tasks.

Returning again to the scenario, possible team configurations (dashed lines) for exchanging tasks are shown in Figure 4. For each driver-agent, the task that is being exchanged is shown. The figure shows that initially, two driver-agents join Ddr-109's team: Asd-102 and Zl-102. These agents announce their respective tasks, which invites additional agents to join. Note that driver-agents Asd-102 and Ddr-109 each participate twice in the extension process.

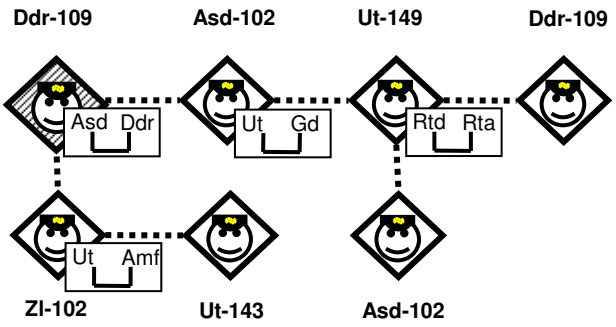


Figure 4: Possible scenario team configurations

Cost function: A cost-function assigns costs to a task exchange based on the status of the driver-agent and the impact of the task exchange. The cost function is strictly increasing and assigns costs to the following elements:

- Extending a schedule past the original end time, and introducing overtime in a schedule;
- Losing meal breaks;
- Replacing stand-by tasks as opposed to regular free time in a schedule;
- Team configuration: Joining as a new team member as opposed to a recurring team member.

Phase 3: Choosing final team configuration: The team extension process is considered complete when a sequence of task exchanges is determined in which all conflicts have been resolved, or any remaining conflicts are sufficiently shifted forward in time to be resolved at a later point in time (re-introduced as new conflicts later). At this point, the recursive team formation process is 'backtracked': Each layer within a team selects the task exchange associated with the lowest cost. In Figure 5, the costs associated with each potential task exchange in the scenario are shown as labels of the edges. Ultimately, a team leader receives an overview of the potential team configurations from each driver-agent in the first team layer. By comparing the costs of the configurations the team leader selects a final team (solid lines).

Phase 4: Finalizing solution: Once a configuration has been selected, the team leader notifies the involved driver-agents that the configuration has been accepted. When all team leaders have determined a suitable solution for their specific conflicts resulting from the disruption, these solutions are presented to the dispatcher.

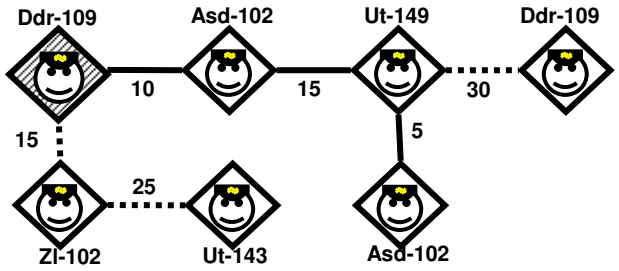


Figure 5: Choosing the final team configuration

2.5 Managing the team formation process

Considering the number of driver-agents involved in the rescheduling process, as well as the high degree of connectivity in the rail network, the number of possible team configurations examined in this manner is very large. In addition, driver-agents are designed to participate in multiple teams and team configurations within these teams to maximize the chance of finding favorable configurations, allowing temporary conflicts. Two mechanisms are applied to manage the dynamic team formation process:

1. Commitment levels: During a task exchange process a driver-agent increases its commitment level to this task exchange. Every increase makes it more difficult for that driver-agent to decommit from the task exchange. In the final commitment level (i.e. 'full commitment') a driver-agent must ensure that any ongoing task exchanges that overlap with the fully committed task exchange are aborted.

2. Task exchange strategies: At several points in the team formation process, driver-agents apply strategic knowledge to determine the best course of action. Although driver-agents are considered to be self-interested with respect to the driver's preferences, these strategies are aimed to guide the team formation process to find solutions that have globally favorable properties, and to dismiss less favorable solutions early on in the solution process. Examples of strategies used by driver-agents are:

- **Cost function**: By assigning different costs to the cost function elements, team configurations with specific properties can be favored in the configuration process. For example, increasing the cost for accepting overtime in a schedule will lead to solutions containing overtime to be dismissed in favor of solutions that modify schedules without introducing overtime.
- **Interest determination strategy**: Determines whether a task exchange is worthwhile before joining a team. A scoreboard mechanism is applied to publish current team scores and inform potential new team members.
- **Decommitment strategy**: Determines how concurrent, overlapping task exchanges are handled.

3 Current Implementation

The described actor-agent based solution is currently being realized: The research system is to be delivered at the end of 2008. This section provides a brief synopsis of our first findings. For the implementation of the agents in the prototype system, the Cougaar [4] agent framework is used. Development of the prototype follows an iterative process, each iteration consisting of adjusting system design and requirements, implementing design-changes, and evaluating system behavior.

In the implementation, interaction with a dispatcher-actor is achieved by means of a GUI which presents schedule representations resembling those of rescheduling tools currently used by dispatchers. Currently, the GUI can also be used by a dispatcher-actor to introduce specific disruption scenarios into the system for testing purposes. In order to run realistic scenarios, a dataset containing a timetable and driver/rolling stock schedules for a full day has been provided by the NS. The current version (September 2008) of the research system is able to find solutions for relatively large disruptions. To illustrate this, results of an example scenario are presented: The scenario consists of a complete blockage between Groningen and Zwolle from 17:00 to 18:00. The number of cancelled train services due to this blockage is 11, which leads to 11 driver-agents to act as team-leaders. Table 1 shows the results of various runs with different driver-agent populations. Some remarks can be made concerning these results:

- The additional spare driver-agent added in run 2 eliminates the overtime generated in run 1. The time needed to find a solution is also reduced.
- The additional driver-agents added in run 5 represent train drivers that are located far from the actual disruption location. Team configurations containing these agents are quickly discarded in the solution process. The calculation time remains the same as in run 4.

	# driver-agents	# task exchanges	total # team members	overtime (min)	calculation time[1]
1	52 (no spare drivers)	20	15	96	5:50
2	53 (1 spare driver)	16	14	0	3:00
3	82 (no spare drivers)	20	14	0	9:10
4	84 (2 spare drivers)	equal to 3.	equal to 3.	equal to 3.	6:40
5	123 (2 spare drivers)	equal to 3.	equal to 3.	equal to 3.	6:40
6	177 (2 spare drivers)	equal to 3.	equal to 3.	equal to 3.	11:30

Table 1: Disruption scenario results

4 Related work

Traditionally, Operations Research approaches are employed in the field of crew rescheduling. Jespersen-Groth et al. present an overview of railway disruption management, including crew rescheduling in [5], describing both the process itself and the directly involved organizations. The authors mention the lack of computerized support for railway disruption management, and a case is made for the use of Operations Research techniques in the disruption management process. Furthermore, the paper presents a comparison with disruption management in the airline industry, in which similar rescheduling goals are distinguished.

Agent-based crew-rescheduling is a relatively new area of research. De Weerdt et al. state in their overview of multi-agent planning [7] that although most researchers recognize the importance of dealing with changing environments, most planning approaches still assume fairly stable worlds. The authors mention contingency planning (plan for all contingencies that might occur) as a traditional approach of handling changes in the environment. As in many situations planning for all possible contingencies is not feasible, the authors argue that so-called *plan repair* approaches are more realistic: Detecting deviations from the original plan through monitoring, and adjust the plan as needed. DesJardins et al. [2] present an overview of approaches in the field of distributed planning. In the paper, approaches are classified according to the properties they share with cooperative distributed planning (emphasis on forming a global (optimal) plan) and negotiated distributed planning (emphasis on satisfying local goals). The authors argue that only recently research in this field has been concerned with coping with dynamic, realistic environments. To cover this emerging work, the authors introduce the *distributed, continual planning* paradigm. This paradigm considers planning to be a dynamic ongoing process combining both planning & execution. The work presented in this paper fits this paradigm, as the crew rescheduling process is performed in real-time and disruptions continuously require agents to revise their schedules to cope with new circumstances.

Mao et al. [6] recognize the need for short-term operational planning and scheduling methods in the domain of airport resource scheduling, and present an agent-based approach based on two coordination mechanisms: decommitment penalties and a Vickrey auction mechanism. The coordination approach used in this paper is based on a combination of similar mechanisms: The driver-agent interaction protocol has auction-like properties (agents report costs (i.e. bid) for taking over tasks), and decommitment penalties are determined based on increasing commitment levels. In literature, coordination approaches based on negotiation concepts are often divided in cooperative and non-cooperative (self-interested) approaches. Although driver-agents in our model can in some respects be considered as self-interested agents (driver preferences are included in the agent's cost function), the agents cooperate to achieve the global goal of resolving disruptions, and agents do not engage in direct competition.

The work presented in this paper can also be viewed in the research context of *personnel scheduling*. Ernst et al. present an overview of application areas, models and algorithms in this area [3]. Application areas they mention include: Transportation systems, call centres, health care systems, and

[1] System configuration: Intel Pentium 3.4 GHz, 2,0 GB RAM

emergency/civic services. Their overview does not include crew-rescheduling approaches in any of these areas. The authors indicate the railway crew scheduling process as a relatively new area of research. Furthermore, the need for more flexible algorithms is recognized, capable of handling changing (work) environments and individual preferences.

5 Future work

A proof-of-concept version of the system described in this paper has been successfully demonstrated in December 2007. Currently, the prototype is extended to include all train drivers and more elaborate disruption scenarios. The effectiveness of the team-based task exchange approach is already showing its first promising results; more thorough analyses are planned to be conducted in the final quarter of 2008 when the research system is linked to real-time disruption information. On the longer term extending the system to other rescheduling tasks, such as the rescheduling of conductors, is foreseen.

Acknowledgments
The authors express their gratitude to NS and Cor Baars (DNV / Cibit) for starting this project. The following D-CIS Lab colleagues provided valuable contributions: Louis Oudhuis, Martijn Broos, Sorin Iacob, and Thomas Hood. The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (http://www.icis.decis.nl/), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The ICIS project is hosted by the D-CIS Lab (http://www.decis.nl/), the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam and the Netherlands Organisation for Applied Scientific Research (TNO).

References

[1] Abbink, E., Fischetti, M., Kroon, L., Timmer, G., and Vromans, M. (2005). Reinventing crew scheduling at Netherlands Railways. In: *Interfaces*, **35**, pp. 393-401

[2] desJardins, M.E., Durfee, E.H., Ortiz, C.L., and Wolverton, M.J. (1999). A Survey of Research in Distributed, Continual Planning, *AI Magazine*, **4**, pp. 13-22

[3] Ernst, A.T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004), Staff Scheduling and Rostering: A Review of Applications, Methods, and Models. In: *Eur. Jnl of Operational Research*, **153**, pp. 3-27.

[4] Helsinger, A., Thome, M., and Wright, T. (2004), Cougaar: A Scalable, Distributed Multi-agent Architecture. In: *Proc. of the Int. Conf. on Systems, Man and Cybernetics*, The Netherlands.

[5] Jespersen-Groth, J., Pothoff, D., Clausen, J., Huisman, D., Kroon, L., Maróti, G., and Nyhave Nielsen, M. (2007), *Disruption Management in Passenger Railway Transportation*. Report EI2007-05, Econometric Institute, Erasmus University Rotterdam (2007), 35 pages. (Submitted to Computers & Operations Research (special issue on disruption management) in January 2007)

[6] Mao, X., ter Mors, A., Roos, and N., Witteveen, C. (2007), Coordinating Competitive Agents in Dynamic Airport Resource Scheduling. In P. Petta, J. P. Mueller, M. Klusch, M. Georgeff (Eds.). *Proc. of the 5th German Conf. on Multiagent System Technologies*, LNAI, Springer Verlag, vol. **4687**, pp. 133-144.

[7] de Weerdt, M., ter Mors, A., and Witteveen, C. (2005), Multi-agent Planning: An introduction to planning and coordination. In: *Handouts of the European Agent Summer School*, pp. 1-32.

[8] Wijngaards, N., Kempen, M., Smit, A., and Nieuwenhuis, K. (2006), Towards Sustained Team Effectiveness. In: Lindemann, G., et al. (Eds.), Selected revised papers from the workshops on *Norms and Institutions for Regulated Multi-Agent Systems* (ANIREM) and *Organizations and Organization Oriented Programming* at AAMAS'05, LNCS, Springer Verlag, vol. **3913**, pp. 33-45.

Rapidly Adapting Game AI

Sander Bakkes Pieter Spronck Jaap van den Herik

Tilburg University / Tilburg Centre for Creative Computing (TiCC)
P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands
{s.bakkes,p.spronck,h.j.vdnherik}@uvt.nl

Abstract

Current approaches to adaptive game AI require either a high quality of utilised domain knowledge, or a large number of adaptation trials. These requirements hamper the goal of rapidly adapting game AI to changing circumstances. In an alternative, novel approach, domain knowledge is gathered automatically by the game AI, and is immediately (i.e., without trials and without resource-intensive learning) utilised to evoke effective behaviour. In this paper we discuss this approach, called 'rapidly adaptive game AI'. We perform experiments that apply the approach in an actual video game. From our results we may conclude that rapidly adaptive game AI provides a strong basis for effectively adapting game AI in actual video games.

1 Introduction

Over the last decades, modern video games have become increasingly realistic with regard to visual and auditory presentation. Unfortunately, game AI has not reached a high degree of realism yet. Game AI is typically based on non-adaptive techniques [18]. A major disadvantage of non-adaptive game AI is that once a weakness is discovered, nothing stops the human player from exploiting the discovery. The disadvantage can be resolved by endowing game AI with adaptive behaviour, i.e., the ability to learn from mistakes. Adaptive game AI can be established by using machine-learning techniques, such as artificial neural networks or evolutionary algorithms. In practice, adaptive game AI in video games is seldom implemented because machine-learning techniques typically require numerous trials to learn effective behaviour. To allow rapid adaptation in games, in this paper we describe a means of adaptation that is inspired by the human capability to solve problems by generalising over a limited number of experiences with a problem domain.

The outline of this paper is as follows. First, we discuss the aspect of entertainment in relation to game AI. Then, we discuss our approach to establish rapidly adaptive game AI. Subsequently, we describe an implementation of rapidly adaptive game AI. Next, we describe the experiments that apply rapidly adaptive game AI in an actual video game, followed by a discussion of the experimental results. Finally, we provide conclusions and describe future work.

2 Entertainment and Game AI

The purpose of a typical video game is to provide entertainment [18, 12]. Of course, the criteria of what makes a game entertaining may depend on who is playing the game. Literature suggests the concept of immersion as a general measure of entertainment [11, 17]. Immersion concerns evoking an immersed feeling with a video game, thereby retaining a player's interest in the game. As such, an entertaining game should at the very least not repel the feeling of immersion from the player [9]. Aesthetical elements of a video game, such as graphics, narrative and rewards, are instrumental in establishing an immersive game-environment. Once established, the game environment needs to uphold some form of *consistency* for the player to remain immersed within it [9]. Taylor [17] argues that a lack of consistency in a game can cause player-immersion breakdowns.

The task for game AI is to control game characters in such a way that behaviour exhibited by the characters is consistent within the game environment. In a realistic game environment, realistic character behaviour

is expected. As a result, game AI that is solely focused on exhibiting the most *challenging* behaviour is not necessarily regarded as realistic. For instance, in a typical first-person shooter (FPS) game it is not realistic if characters controlled by game AI aim with an accuracy of one hundred per cent. Game AI for shooter games, in practice, is designed to make intentional mistakes, such as warning the player of an opponent character's whereabouts by intentionally missing the first shot [10].

Consistency of computer-controlled characters with a game environment is often established with tricks and cheats. For instance, in the game HALF-LIFE, tricks were used to establish the illusion of collaborative teamwork [9], causing human players to assume intelligence where none existed [10]. While it is true that tricks and cheats may be required to uphold consistency of the game environment, they often are implemented only to compensate for the lack of sophistication in game AI [4]. In practice, game AI in most complex games still is not consistent with the game environment, and exhibits what has been called 'artificial stupidity' [10] rather than artificial intelligence. To increase game consistency, and thus the entertainment value of a video game, we agree with Buro and Furtak [4] that researchers should foremost strive to create the most optimally playing game AI possible. In complex video-games, such as real-time strategy (RTS) games, near-optimal game AI is seen as the only way to obtain consistency of the game environment [9]. Once near-optimal game AI is established, difficulty-scaling techniques can be applied to downgrade the playing-strength of game AI, to ensure that a suitable challenge is created for the player [15].

3 Approach

For game AI to be consistent with the game environment in which it is situated, it needs the ability to adapt adequately to changing circumstances. Game AI with this ability is called 'adaptive game AI'. Typically, adaptive game AI is implemented for performing adaptation of the game AI in an online and computer-controlled fashion. Improved behaviour is established by continuously making (small) adaptations to the game AI. To adapt to circumstances in the current game, the adaptation process typically is based only on observations of *current* gameplay. This approach to adaptive game AI may be used to improve significantly the quality of game AI by endowing it with the capability of adapting its behaviour while the game is in progress. For instance, the approach has been successfully applied to simple video games [5, 8], and to complex video games [15]. However, this appproach to adaptive game AI requires either (1) a high quality of the utilised domain knowledge, or (2) a large number of adaptation trials. These two requirements hamper the goal of achieving *rapidly* adaptive game AI.

To achieve rapidly adaptive game AI, we propose an alternative, novel approach to adaptive game AI that comes without the hampering requirements of typical adaptive game AI. The approach is coined 'rapidly adaptive game AI'. We define rapidly adaptive game AI as an approach to game AI where domain knowledge is gathered automatically by the game AI, and is immediately (i.e., without trials and without resource-intensive learning) utilised to evoke effective behaviour. The approach, illustrated in Figure 1, implements a direct feedback loop for control of characters operating in the game environment. The behaviour of a game

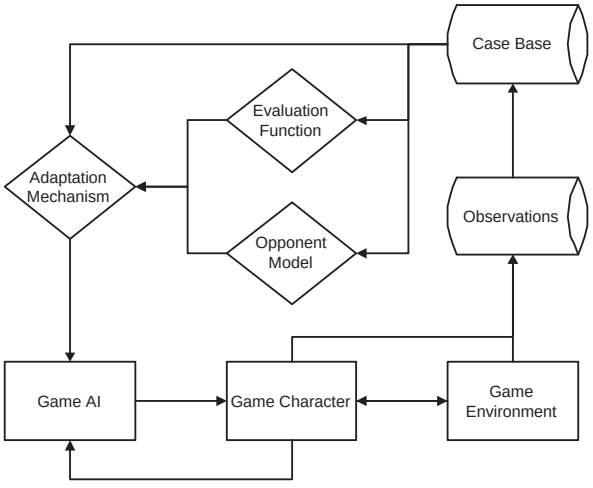


Figure 1: Rapidly adaptive game AI (see text for details).

character is determined by the game AI. Each game character feeds the game AI with data on its current situation, and with the observed results of its actions. The game AI adapts by processing the observed results, and generates actions in response to the character's current situation. An adaptation mechanism is incorporated to determine how to best adapt the game AI. For instance, reinforcement learning may be applied to assign rewards and penalties to certain behaviour exhibited by the game AI.

For rapid adaption, the feedback loop is extended by (1) explicitly processing observations from the game AI, and (2) allowing the use of game-environment attributes which are not directly observed by the game character (e.g., observations of team-mates). Inspired by the case-based reasoning paradigm, the approach collects character observations and game environment observations, and extracts from those a case base. The case base contains all observations relevant for the adaptive game AI, without redundancies, time-stamped, and structured in a standard format for rapid access. To rapidly adapt to circumstances in the current game, the adaptation process is based on domain knowledge drawn from observations of a *multitude* of games. The domain knowledge gathered in a case base is typically used to extract models of game behaviour, but can also directly be utilised to adapt the AI to game circumstances. In our proposal of rapidly adaptive game AI, the case base is used to extract an evaluation function and opponent models. Subsequently, the evaluation function and opponent models are incorporated in an adaptation mechanism that directly utilises the gathered cases.

The approach to rapidly adaptive AI is inspired by the human capability to reason reliably on a preferred course of action with only a few observations on the problem domain. Following from the complexity of modern video games, game observations should, for effective and rapid use, (1) be represented in such a way that stored cases can be reused for previously unconsidered situations, and (2) be compactly stored in terms of the amount of retrievable cases [1]. As far as we know, rapidly adaptive game AI has not yet been implemented in an actual video game.

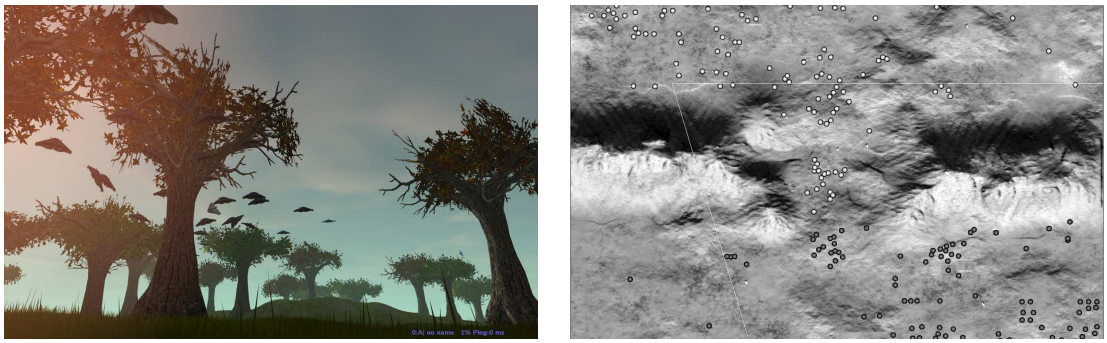
4 Implementation

This section discusses our proposed implementation of rapidly adaptive game AI. In the present research we use SPRING [7], illustrated in Figure 2(a), which is a typical and open-source RTS game. In SPRING, as in most RTS games, a player needs to gather resources for the construction of units and buildings. The aim of the game is to defeat an enemy army in a real-time battle. A SPRING game is won by the player who first destroys the opponent's 'Commander' unit.

We subsequently discuss (1) the evaluation function, (2) the established opponent models, and (3) an adaptation mechanism inspired by the case-based reasoning paradigm.

4.1 Evaluation Function

To exhibit behaviour consistent with the game environment presented by modern video games, game AI needs the ability to accurately assess the current situation. This requires an appropriate evaluation function.



(a) (b)

Figure 2: Two screenshots of the SPRING game environment. In the first screenshot, airplane units are flying over the terrain. In the second screenshot, an overview is presented of two game AI's pitted against each other on the map 'SmallDivide'.

The high complexity of modern video games makes the task to generate such an evaluation function for game AI a difficult one.

Previous research discussed an approach to automatically generate an evaluation function for game AI in RTS games [3]. The approach incorporates TD-learning [16] to learn unit-type weights for the evaluation function. Our evaluation function for the game's state is denoted by

$$v(p) = w_p v_1 + (1 - w_p)v_2 \tag{1}$$

where $w_p \in [0 \ldots 1]$ is a free parameter to determine the weight of each term v_n of the evaluation function, and $p \in \mathbb{N}$ is a parameter that represents the current *phase of the game*. In our experiments, we defined five game phases and used two evaluative terms, the term v_1 that represents the material strength and the term v_2 that represents the Commander safety. Our experimental results showed that just before the game's end, the established evaluation function is able to predict correctly the outcome of the game with an accuracy that approaches one hundred per cent. Additionally, the evaluation function predicts ultimate wins and losses accurately before half of the game is played. From these results, we concluded that the established evaluation function effectively predicts the outcome of a SPRING game. Therefore, we incorporated the established evaluation function in the implementation of our rapidly adaptive game AI.

4.2 Opponent Models

An additional feature of consistent behaviour in game AI is the ability to recognise the strategy of the opponent player. This is known as opponent modeling. In the current experiment, we will not yet incorporate opponent modeling, for first the effectiveness of the adaptation mechanism will be established in dedicated experimentation.

However, previous research already discussed a successful approach for opponent modeling in RTS games [13]. In the approach, a hierarchical opponent model of the opponent's strategy is established. The models are so-called fuzzy models [19] that incorporate the principle of discounted rewards for emphasising recent events more than earlier events. The top-level of the hierarchy can classify the general play style of the opponent. The bottom-level of the hierarchy can classify strategies that further define behavioural characteristics of the opponent.

Experimental results showed that the general play style can accurately be classified by the top-level of the hierarchy. Additionally, experimental results obtained with the bottom-level of the hierarchy showed that in early stages of the game it is difficult to obtain accurate classifications. In later stages of the game, however, the bottom-level of the hierarchy will accurately predict the opponent's specific strategy. From these results, it was concluded that the established approach for opponent modeling in RTS games can be successfully used to classify the strategy of the opponent while the game is still in progress.

4.3 Adaptation Mechanism

In our approach, domain knowledge collected in a case base is utilised for adapting game AI. To generalise over observations with the problem domain, the adaptation mechanism incorporates a means to index collected games, and performs a clustering of observations. For action selection, a similarity matching is performed that considers six experimentally determined features. The adaptation process is algorithmically described below.

```
//Offline processing
A1. Game indexing: to calculate indexes for all stored games.
A2. Clustering of observations: to group together similar observations.

//Online action selection
B1. Use game indexes to select the N most similar games.
B2. Of the selected N games, select the M games that best satisfy the goal criterion.
B3. Of the selected M games, select the most similar observation.
B4. Perform the action stored for the selected observation.
```

Game indexing (A1): We define a game's index as a vector of fitness values, containing one entry for each time step. These fitness values represent the desirability of all observed game states. To calculate the fitness value of an observed game state, we use the previously established evaluation function (denoted in Equation 1). Game indexing is supportive for later action selection, and as it is a computationally-expensive procedure, it is performed offline.

Clustering of observations (A2): As an initial means to cluster similar observations, we apply the standard k-means clustering algorithm [6]. The metric that expresses an observation's position in the cluster space is comprised of a weighted sum of the six observational features that also are applied for similarity matching. Clustering of observations is supportive for later action selection, and as it is a computationally-expensive procedure, it is performed offline.

Similarity matching (A2 and B3): To compare a given observation with another, we define six observational features, namely (1) phase of the game, (2) material strength, (3) commander safety, (4) positional footprint, (5) economical strength, and (6) unit count. Similarity is defined by a weighted sum of the absolute *difference* in features values.[1] As observations are clustered, calculating the similarity between observations is relatively computationally-inexpensive. This is important, as similarity matching must be performed online.

Action selection (B1-B4): Using the established game indexes, we select the N games with the smallest accumulated fitness difference with the current game, up until the current observation. Subsequently, of the selected N games, we perform the game action of the most similar observation of the M games that satisfy a particular goal criterion. The goal criterium can be any metric to represent preferred behaviour. For instance, a preferred fitness value of 0 can represent challenging gameplay, as this implies that players are equally matched. Naturally, we have to consider that performing actions associated to similar observations may not yield the same outcome when applied to the current state. Therefore, to estimate the effect of performing the retrieved game action, we straightforwardly compensate for the difference in metric value between the current and the selected observation.

5 Experiments

This section discusses experiments that test our implementation of rapidly adaptive game AI. We first describe the experimental setup and the performance evaluation, and then the experimental results.

5.1 Experimental Setup

To test our implementation we start collecting observations of games where two game AIs are posed against each other. Multiple SPRING game AIs are available. We found one game AI which was open source, which we labelled 'AAI' [14]. We enhanced this game AI with the ability to collect game observations in a case base, and the ability to disregard radar visibility so that perfect information on the environment was available. As opposing game AI, we used AAI itself. We found 27 parameters that define the strategic behaviour of the game AI.[2] To simulate different players competing with different players, for each game the strategic parameters of both players are pseudo-randomised. The data collection process was as follows. During each game, game observations were collected every 127 game cycles, which corresponds to the update frequency of AAI. With the SPRING game operating at 30 game cycles per second, this resulted in game observations being collected every 4.233 seconds. Of each game observation, the position and unit-type of every unit is abstracted. The games were played on the map 'SmallDivide', illustrated in Figure 2(b), which is a symmetrical map without water areas. All games were played under identical starting conditions. Accordingly, a case base was built from 130552 observations of 200 games, resulting in a total of 392 MB of uncompressed observational data. Note that approaches are available for offline data compression and subsequent online data decompression [2], but these lie outside the scope of the present research.

For clustering of observations, k is set to ten per cent of the total number of observations. For action selection, the $N = 50$ games with the smallest fitness difference with the current game are selected. Subsequently, the game action of the most similar observation in the $M = 5$ games that best satisfy a defined goal criterion, is selected for direct execution. The game action is expressed by the configuration of the 27 parameters of strategic behaviour. Action selection is performed in the beginning of the game, and at every phase transition.

[1] The weights for both clustering of observations and similarity matching are as follows: $(1 + phase\_of\_the\_game) * ((0.5 * unit\_count) + material\_strength + commander\_safety + positional\_footprint + economical\_strength)$.

[2] Three examples of these parameters are AIRCRAFT_RATE (determines how many airplane units the AI will build), MAX_MEX_DEFENCE_DISTANCE (maximum distance to base where the AI defends metal extractors), and MAX_SCOUTS (maximum of units scouting at the same time). The authors happily provide a full list of the parameters on request.

	Trial runs	Goal achieved	Goal achieved (%)
Goal(win)	14	12	86%
Goal(lose)	15	12	80%

Table 1: Effectiveness of rapidly adaptive game AI.

	Average	Standard deviation
Time to uphold tie	32 min.	12 min.
Variance in fitness value	0.16	1.95

Table 2: Rapidly adaptive game AI applied for upholding a tie.

5.2 Performance Evaluation

To evaluate the performance of the rapidly adaptive game AI, we determine to what extent it is capable of adapting effectively when in competition with the original AAI game AI. We define three goals for adaptation, namely (1) winning the game (positive fitness value), (2) losing the game (negative fitness value), and (3) upholding a tie (fitness value of 0, with a fitness difference of at most 10). To measure how well the rapidly adaptive game AI is able to maintain a fitness value of 0, the variance in fitness value is calculated. A low variance implies that the rapidly adaptive game AI has the ability to consistently maintain a predefined fitness value. All experimental trials are repeated 15 times, except the trial to test the rapidly adaptive game AI, which due to a game-engine crash was repeated 14 times.

5.3 Results

Table 1 and Table 2 give an overview of the results of the experiments performed in the SPRING game. Figure 3 displays the obtained fitness value as a function over time of three typical experimental runs. The results reveal that the rapidly adaptive game AI can effectively obtain a victory (86% of the experimental runs), and can effectively lose the game when this is desired (80% of the experimental runs). Subsequently, the results reveal that the rapidly adaptive game AI is capable of upholding a tie for a relatively long time (32 minutes on average), while at the same time maintaining a relatively low variance in the fitness value that is strived for. From these results, we may conclude that rapidly adaptive game AI can be used for effectively adapting game AI in an actual video game.

6 Discussion

In the experiments that test our implementation of rapidly adaptive game AI, we observed that the game AI was not always able to achieve the set goal. A first explanation is that our implementation performs action selection only when a transition in game phase is detected. Though this setup is effective for most games, more moments of action selection may be needed when circumstances are changing rapidly. A second explanation is that our case base, built from 200 games, may still not contain an adequate amount of relevant observations. As rapidly adaptive game AI can be expected to be applied in the playtesting phase of game development, and predictably in multi-player games, the case base in practical applications is expected to grow rapidly to contain a multitude of relevant observations. A last explanation is that eventual outliers cannot be avoided due to the inherent randomness that is typical to video games. For instance, in the SPRING game, the most powerful unit is able to destroy a Commander unit with a single shot. Should the Commander be destroyed in such a way, the question would arise if this was due to bad luck, or due to an effective strategy of the opponent. For game AI to be accepted as effective players, one could argue, recalling the previously mentioned need for consistent AI behaviour, that game AI should not force a situation that may be regarded as the result of lucky circumstances.

We found the rapidly adaptive game AI to be able to uphold a tie for a relatively long time, while at the same time maintaining a relatively low variance in the fitness value that is strived for. This ability may be regarded as a straightforward form of difficulty scaling. If a metric can be established that represents the preferred level of challenge for the human player, then in theory the rapidly adaptive game AI would be capable of scaling the difficulty level to the human player. Such a capability provides an interesting challenge for future research.

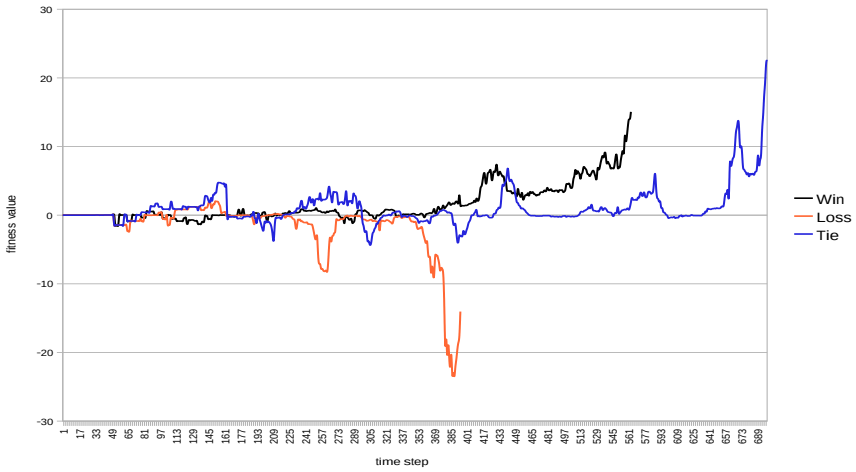


Figure 3: Obtained fitness values as a function over time. The figure displays a typical experimental result of (1) the rapidly adaptive game AI set to win the game, (2) the rapidly adaptive game AI set to lose the game, and (3) the rapidly adaptive game AI set to uphold a tie.

7 Conclusions and Future Work

In this paper we discussed an approach to establish rapidly-adaptive game AI. In the approach, domain knowledge is gathered automatically by the game AI, and is immediately (i.e., without trials and without resource-intensive learning) utilised to evoke effective behaviour. In our implementation of the approach, game observations are collected in a case base. Subsequently, the case base is used to abstract an evaluation function and opponent models, and gathered cases are directly utilised by an adaptation mechanism. Results of experiments that test the approach in the SPRING game show that rapidly-adaptive game AI can effectively obtain a victory, can effectively lose the game when this is desired, and is capable of upholding a tie for a relatively long time. From these results, we may conclude that the established rapidly adaptive game AI provides a strong basis for effectively adapting game AI in actual video games.

For future work, we will extend the established rapidly-adaptive game AI with a means to scale the difficulty level to the human player. Subsequently, we will investigate if our approach to rapidly adapting game AI can be improved by incorporating opponent models.

Acknowledgements. This research is funded by a grant from the Netherlands Organization for Scientific Research (NWO grant No 612.066.406) and is performed in the framework of the ROLEC project.

References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), March 1994.

[2] Samir Abou-Samra, Claude Comair, Robert Champagne, Sun Tjen Fam, Prasanna Ghali, Stephen Lee, Jun Pan, and Xin Li. Data compression/decompression based on pattern and symbol run length encoding for use in a portable handheld video game system. US Patent 6416410, 2002.

[3] Sander Bakkes and Pieter Spronck. *AI Game Programming Wisdom 4*, chapter Automatically Generating Score Functions for Strategy Games, pages 647–658. Charles River Media, Hingham, MA., U.S.A., 2008.

[4] Michael Buro and Timothy M. Furtak. RTS games and real-time AI research. In *Proceedings of the BRIMS Conference*. Arlington VA, 2004.

[5] Pedro Demasi and Adriano J. de O. Cruz. Online coevolution for action games. *International Journal of Intelligent Games and Simulation*, 2(3):80–88, 2002.

[6] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[7] Stefan Johansson, Jelmer Cnossen, and Tomaz Kunaver. Spring game engine. http://spring.clan-sy.com/, 2007.

[8] S Johnson. *AI Game Programming Wisdom 2*, chapter Adaptive AI: A Practical Example, pages 639–647. Charles River Media, Inc., Hingham, MA, 2004.

[9] Ronni Laursen and Daniel Nielsen. Investigating small scale combat situations in real-time-strategy computer games. Master's thesis, Department of computer science, University of Aarhus, Denmark, 2005.

[10] L. Liden. *AI Game Programming Wisdom 2*, chapter Artificial Stupidity: The Art of Making Intentional Mistakes, pages 41–48. Charles River Media, Inc., Hingham, MA, 2004.

[11] Lev Manovich. *The Language of New Media*. The MIT Press, Cambridge, Massachusetts, U.S.A., 2002.

[12] Alexander Nareyek. AI in computer games. *ACM Queue*, 1(10):58–65, 2004.

[13] Frederik Schadd, Sander Bakkes, and Pieter Spronck. Opponent modeling in real-time strategy games. In Marco Roccetti, editor, *Proceedings of the GAME-ON 2007*, pages 61–68, 2007.

[14] Alexander Seizinger. AI:AAI. Creator of the game AI 'AAI', http://spring.clan-sy.com/wiki/AI:AAI, 2006.

[15] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, and Eric Postma. Adaptive game AI with dynamic scripting. *Machine Learning*, 63(3):217–248, 2006.

[16] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[17] Laurie N. Taylor. Video games: Perspective, point-of-view, and immersion. Masters thesis, Graduate Art School, University of Florida, U.S.A., 2002.

[18] Paul Tozour. *AI Game Programming Wisdom (ed. Rabin, S.)*, chapter The Perils of AI Scripting, pages 541–547. Charles River Media, 2002.

[19] Michael Zarozinski. *AI Game Programming Wisdom*, chapter An Open-Fuzzy Logic Library, pages 90–101. Charles River Media, 2002.

Adaptive Intelligence for Turn-based Strategy Games

Maurice Bergsma Pieter Spronck

Tilburg University / Tilburg centre for Creative Computing
P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands

Abstract

Computer games are an increasingly popular form of entertainment. Typically, the quality of AI opponents in computer games leaves a lot to be desired, which poses many attractive challenges for AI researchers. In this respect, *Turn-based Strategy* (TBS) games are of particular interest. These games are focussed on high-level decision making, rather than low-level behavioural actions. Moreover, they allow the players sufficient time to consider their moves. For efficiently designing a TBS AI, in this paper we propose a game AI architecture named ADAPTA (Allocation and Decomposition Architecture for Performing Tactical AI). It is based on task decomposition using asset allocation, and promotes the use of machine learning techniques. In our research we concentrated on one of the subtasks for the ADAPTA architecture, namely the Extermination module, which is responsible for combat behaviour. Our experiments show that ADAPTA can successfully learn to outperform static opponents. It is also capable of generating AIs which defeat a variety of static tactics simultaneously.

1 Introduction

The present research is concerned with artificial intelligence (AI) for turn-based strategy (TBS) games. The AI for TBS games offers many challenges, such as resource management, planning, and decision making under uncertainty, which a computer player must be able to master in order to provide competitive play. Our goal is to create an effective turn-based strategy computer player. To accomplish this, we employ learning techniques to generate the computer player's behaviour automatically. For this research we are most interested in the '4X' subgenre of TBS games. 4X stands for the primary goals of this type of game, namely Exploration, Expansion, Exploitation, and Extermination.

As TBS games are in many ways related to real-time strategy (RTS) games, the research challenges of these domains are rather similar. Buro and Furtak [2] define seven research challenges for RTS games, six of which are also relevant to the TBS domain. These are (1) Adversarial planning, (2) Decision making under uncertainty, (3) Spatial reasoning, (4) Resource management, (5) Collaboration, and (6) Adaptivity. While these six research challenges are similar for both genres, TBS offers more depth in some of them, such as collaboration and resource management. Comparing both genres, one can observe that, usually, RTS games rely more on fast, exciting action, while TBS games rely more on strategic thinking and careful deliberation. As far as research into these domains is concerned, this means that RTS game research is more dependent on the development of time-efficient algorithms, while TBS games require more intelligent AI techniques.

The outline of this paper is as follows. In Section 2 we provide a definition of the TBS game used for this research. Section 3 explains the AI architecture designed for this game. In Section 4 we describe how spatial reasoning is used in our AI, followed by an overview of the learning algorithm used to generate the behaviour of our AI in Section 5. The experimental setup is explained, and the results are discussed in Sections 6 and 7. Section 8 provides conclusions.

2 Game Definition

For this research, we have developed our own game definition, based on Nintendo's ADVANCE WARS, which is a relatively simple TBS game that still supports most of the major features of the genre. Each of the 4 X's

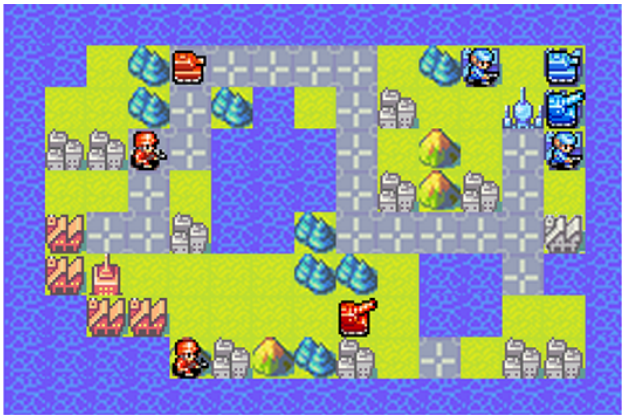


Figure 1: A screenshot of the TBS game under research.

(Exploration, Expansion, Exploitation, and Extermination) are represented in some form in the environment. We named our version SIMPLE WARS. Below, the rules of SIMPLE WARS are explained.

The game takes place on a two-dimensional, tile-based map, as shown in Figure 1. A tile is of a predefined type, such as *Road*, *Mountain*, or *River*. Each type has its own set of parameters. These parameters define the characteristics of the tile, such as the number of moves that it takes for a unit to move over it, or the defense bonus that a unit receives while standing on it. By default, movement in the game can occur either horizontally or vertically, but not diagonally. A defense bonus lowers the amount of damage that the unit receives in combat.

Some tiles contain a base, for example a *City*, which provides a certain amount of resources to the player that controls the base, or a *Factory*, which can produce one unit each turn. Each unit requires a specific amount of resources to be built. A newly created unit is placed on the same tile as the *Factory* that produced it. This unit is unable to move in the turn it was created.

As is the case with bases and tiles, units come in several types. Each type has different values for its parameters, which include (1) the starting amount of health points for the unit, (2) the number of moves it can make per turn, (3) the amount of damage it can do to each type of unit, and (4) the actions the unit can perform. Because a unit in the game actually represents a squad of units, and damaging the unit represents destroying a part of the squad, the effectiveness of actions such as *Attack* (attacking another unit) is tied to the number of health points that remains for the unit. Additionally, the effectiveness of an *Attack* depends on the types of units involved.

The three types of units in this game are *Infantry*, *Tank*, and *Anti-Tank* units. Of these, only the *Infantry* unit is capable of moving over *Mountain* tiles. None of these units is able to move over *Sea* tiles. The units follow a rock-paper-scissors approach, which means that each of the three types is, where doing damage to the other two unit types is concerned, stronger than one and weaker than the other (Infantry defeats Anti-Tank, Anti-Tank defeats Tank, and Tank defeats Infantry).

At every turn, a player can build a single unit at each *Factory* under control, and perform (1) a *Move* and (2) an *Attack* or other action for each unit under control. An *Attack* action can be performed without moving, but a unit cannot move after performing this action. A tile can only contain a single unit at any time. This implies that a *Factory* is unable to produce a new unit whenever another unit is located at the same tile as this *Factory*. Additionally, a moving unit is able to pass through a tile occupied by a friendly unit, but not through a tile occupied by an enemy unit.

3 The ADAPTA Architecture

In many commercial strategy games, the AI is implemented using scripts, which quickly become difficult to balance or adapt. In order to keep the size of the AI manageable, the complex task in the script can be decomposed into several subtasks, which operate independently of each other, and concentrate each on performing a specific part of the complex task, without having to take any of the other subtasks into account. Examples of possible subtasks include resource gathering and combat.

While the goals of the subtasks are independent, they all share the same environment, namely the game

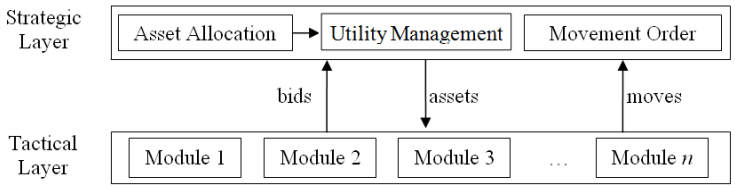


Figure 2: The ADAPTA Game AI architecture.

world. Moreover, they need to share the finite number of *assets* (resources and game objects) that are available. By nature, a subtask only sees a part of the big picture, and is not concerned with the overall path to victory. Therefore, separate modules are required to keep track of the AI's goals, to determine which goals (and therefore, which subtasks) have priority at a point in the game, and to allocate control over the available assets to the different subtasks. Because these modules keep track of the overall strategy, they are called the strategic modules. The subtasks are called tactical modules, as they are each responsible for one type of tactics. In combination, all these modules make up the ADAPTA architecture, depicted in Figure 2.

In the ADAPTA architecture, the strategic AI acts as an arbitrator between the different tactical modules. The Asset Allocation Module decides which tactical module gets control over which assets. This is achieved through auctioning. Tactical modules generate bids, which consist of one or more assets that a module wants to use, and a utility value that the module assigns to these assets (e.g., it assigns a high utility value to assets which it considers to be very useful for achieving its goals). The Asset Allocation Module uses these bids to find the allocation which maximises 'social welfare.' Bids are generated by the various tactical modules, which are not concerned with the bids of competing modules. Therefore, the strategic layer contains a Utility Management Module which weighs each bid's utility value according to the tactical module that generated it and the overall goal of the game. After the assets have been allocated, the Movement Order Module decides in which order the actions generated by the tactical modules are executed.

A tactical module is required to perform three tasks, namely (1) Bid Generation, (2) Utility Calculation, and (3) Action Generation.

1. A bid generator is responsible for submitting a set of bids to the Asset Allocation module. These bids should represent the optimal (or near optimal) actions for a subset of the available assets, according to a certain tactic. The ADAPTA architecture does not impose a specific method to generate the set of bids. Rather, the approach is entirely up to the designer of the module.

2. A utility function calculates a numerical utility value given a certain set of assets. This utility is used by the strategic layer to determine asset allocation. Utilities should provide an accurate measurement of the relative effectiveness of a certain tactic, compared to other tactics generated by the same module.

3. After all tactical modules have submitted their bids and the asset allocation is performed, assets are assigned to the tactical modules that submitted the winning bids. The modules generate game actions for the assets assigned to them. These actions are submitted (along with the utility of the associated bid) to the Movement Module in the strategic layer, which executes them in a certain order.

4 Spatial Reasoning

In this section we will take a closer look at the tactical layer of the ADAPTA architecture. As defined in the previous section, a tactical layer can contain any number of tactical modules, and a module can be responsible for any task. This chapter will focus on the creation of a single tactical module, which will serve as an example of how tactical modules fit into the ADAPTA architecture. This example module is named the Extermination module, as it is responsible for handling combat between units. In our environment, this amounts to assigning Move and Attack actions to each unit under the command of an AI.

In our research, we use influence maps [5, 6], to determine the optimal tile to perform an action. This means that the higher the influence is at a tile, the more preferable it is for a certain unit to move to this tile. For TBS games, an influence map is typically calculated just before making a decision, so that it will, due to the turn-based nature, always reflect the current game state.

The influence map is calculated separately for each tile, as follows:

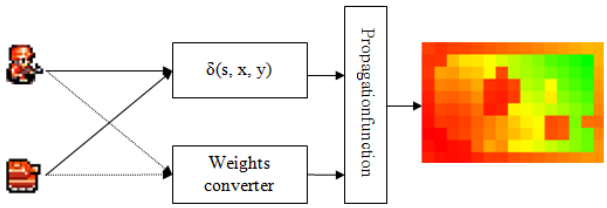


Figure 3: The calculation of an influence map.

$$I(x,y) = \sum_{o \in O} p(w(o), \delta(o,x,y)),$$

where O is the set of all objects used for this influence map, $p(W,d)$ is a *propagation function* of weight vector W (values for attributes of the object which are used by the propagation function; often just one value for each object) and distance d, $w(o)$ converts object o into a weight vector, and $\delta(o,x,y)$ is a distance function calculating the distance from object o to tile (x,y). Simply put, the propagation function will ensure that the influence of an object is higher for tiles that are close to the object, and lower further away from the object. A graphical representation of the calculation of an influence map is shown in Figure 3. In this case, the influence map concerns moveable unit types, and may represent a concept such as 'offensive power.'

The behaviour of an influence map is defined by the two aforementioned functions, (1) the distance function δ and (2) the propagation function p. The distance function can be a general distance metric, such as Euclidean or Manhattan distance, or a domain-specific function, such as the number of steps that a unit needs to get to the target tile. The propagation function defines the influence that an object has on each tile. It is a function of the distance between the game object and current tile. Note that, for the propagation function, values below zero are set to zero, because the object has no influence in the corresponding tile rather than negative influence.

Because different types of game objects have different effects on the game, the calculation of influences should be performed differently for each type. Additionally, by summing both positive and negative influences for different types of objects in a single influence map, information may be lost. For these reasons, multiple influence maps should be maintained for different types of objects. These influence maps can then be analysed on their own as well as combined with others by a *layering algorithm*. A layering algorithm is a function that combines the contents of a number of influence maps and combines them into a single, new influence map. Figure 4 is an extension of Figure 3 which illustrates the process of layering two different influence maps, namely the one from Figure 3, and one concerning stationary units, which may represent a concept such as 'production capacity.'.

In our research, the layering algorithm is a neural network, which uses all influence maps that were the result of propagation functions as inputs. As output, it generates two influence maps simultaneously, one of

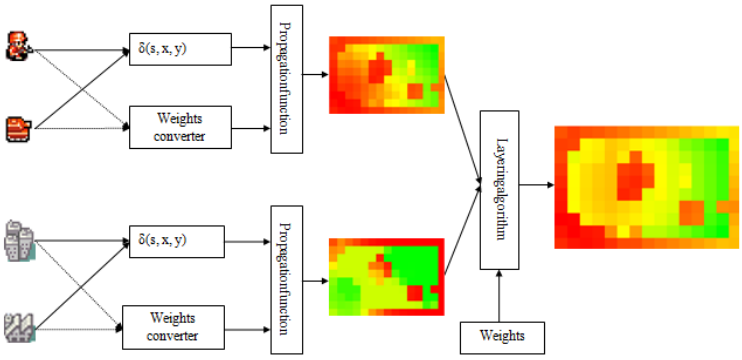


Figure 4: Layering two influence maps.

which indicates the preferability of moving to each tile, and the other doing the same for attacking each tile. In this way, the task of the Extermination module is reduced to selecting the tiles with the highest influence from all possible targets for the Move and Attack actions.

5 Adaptivity

The behaviour of an AI using the spatial reasoning approach described in the previous section depends on the choices for the propagation functions and layering algorithm, as well as their weights. To generate effective behaviour for such an AI, we use a learning algorithm. In order to keep the size of the search space manageable, we limit ourselves to generating the weights, while the choices of all functions remain fixed.

The learning algorithm we used is an evolutionary algorithm (EA) [3, 7]. In our approach, candidate solutions, or *individuals*, are represented by adding the weight strings of each neural network together, forming a single string of numbers. At each generation, the effectiveness (or *fitness*) of each of the individuals must be determined according to a certain *fitness measure*. Because the individuals define the behaviour of a game AI, their fitness can be determined by letting these AIs play the game that they are intended for. In order to attain consistent fitness values, all individuals are made to play against the same opponent(s). An individual's fitness is defined as

$$F_i = \frac{\sum_j R_{ij} + \sum_j R_{ji}}{2 \cdot |R_i|,}$$

where R_{ij} is the numerical value resulting from a game between players i and j, where player i has starting position 1 and player j has starting position 2. The genetic operators chosen for the EA are capable of operating on neural networks. The operators are able to change single weights, entire nodes, or complete networks. A description of each of the operators, as well as a detailed overview of the EA used, is given by Bergsma [1].

6 Experiment 1: Iterative Learning

For our experiments, we implemented a research environment in which the game SIMPLE WARS can be played between different AIs.

It is impractical for the ADAPTA AI to learn against human players. We decided to determine the ability of the ADAPTA AI to defeat a certain tactic by letting it play against other AI implementations, such as scripts. We implemented a simple rush tactic to serve as an initial opponent. This tactic always makes its available units move towards the nearest enemy unit, and attacks whenever possible. In practice, for many commercial RTS games, rush tactics are considered to be a very strong, sometimes optimal approach to win the game [4].

Because the goal of this experiment is to train the Exterminate module, the game rules are focussed on combat. The goal of the game is to destroy 10 enemy units. This is also the total amount of units a player is allowed to build during the game. Because the game is played on a small map, both players are only allowed to control 4 units at any time. Both players start off without any units, but with 4 Factories, each of which is able to build all three unit types. Aside from the 10-unit limitation, both players are given sufficient resources to build 10 units at the start of the game.

Our goal is to improve iteratively the performance and behaviour of the ADAPTA AI, as well as assessing to which extent the results of the learning algorithm generalise over different opponents. Therefore, after the ADAPTA AI has learned to convincingly defeat the initial opponent, we started a new run with the learned AI now as the new opponent. Again, when this opponent was convincingly defeated, we replaced it with the newly generated AI. This process was repeated until ten new AIs were generated.

Table 1 lists the final minimum, average, maximum, and opponent fitness values for each iteration. The Rush AI is used as the opponent in the first iteration, the AI generated against the Rush AI is used as the opponent in the second iteration, etc.

From Table 1 we conclude that, at each iteration, the learning algorithm is successful in generating a solution which outperforms its opponent. This can be derived from the fact that the maximum fitness for each iteration is positive. To compare the individual results, all eleven AIs (the Rush AI and the ten generated AIs) were made to play against each other. For each of these AIs, their fitnesses against emphevery opponent are averaged in Figure 5 (this is different from the average listed in Table 1, which is the average fitness achieved against only its training opponent). Here, iteration number 0 represents the Rush AI.

Iter.	Minimum	Average	Maximum	Opponent
1	-0.293	0.129	0.473	-0.129
2	-0.275	0.059	0.326	-0.059
3	-0.060	0.155	0.486	-0.155
4	-0.416	-0.027	0.286	0.027
5	-0.021	0.318	0.591	-0.318
6	-0.252	0.141	0.500	-0.141
7	-0.533	0.057	0.357	-0.057
8	-0.022	0.025	0.302	-0.025
9	-0.425	0.053	0.300	-0.053
10	-0.212	0.111	0.457	-0.111

Table 1: Final fitness values for Experiment 1.

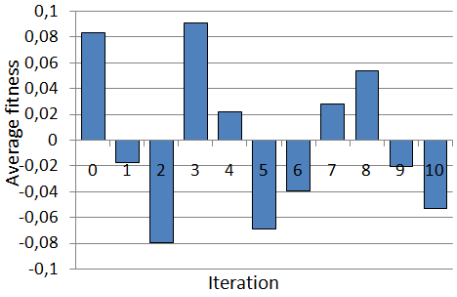


Figure 5: Average fitnesses of all AIs.

Figure 5 shows clearly that the overall performance of the AIs does not increase with each iteration. Even though each AI outperforms the AI from the previous iteration, the average fitnesses may go up or down. This fact suggests that the results do not generalise well over the opponents from previous iterations.

For a better understanding of these results, we considered the types of tactics that the generated AIs use. We found that the AIs can be divided into three categories, each of which uses simple tactics.

1. The Defence tactic. This tactic keeps units close to each other at the base, and only attacks when provoked. It is used by AIs #1 and #5.

2. The Base Offence tactic. This tactic entails rushing the enemy bases with all units, and attacking enemy units whenever possible. It is used by AIs #2, #4, and #7.

3. The Unit Offence tactic. This tactic is similar to the Base Offence tactic, but it moves towards enemy units instead of bases. It is used by AIs #3, #6, #8, #9, and #10.

Figure 6 shows the performances for each of these types of tactics against each of the others, as well as against the Rush tactic. Each of the four graphs contains the average fitness values for the corresponding tactic against each of the four tactics. From this figure we can observe the following. The Rush AI outperforms the Base Offence and Unit Offence tactics, but is outperformed by the Defence tactic. In turn, the Defence tactic is outperformed by the Base and Unit Offence tactics. This implies that none of the tactics dominates the others. It seems that, as is the case with the units, a rock-paper-scissors relationship exists at the level of strategies.

As a side note, we add that no Fog of War (FoW) was used for these experiments. While experiments with FoW were performed, their results were similar to the results described here. A possible explanation for this is that, for this environment, FoW has little effect on the game. Details concerning the FoW experiments are given by Bergsma [1].

7 Experiment 2: Multi-Objective Learning

Because each generated AI's performance was strongest against their direct opponent, it is expected that an AI which is learned against multiple opponents will be relatively strong against each of them. If each of

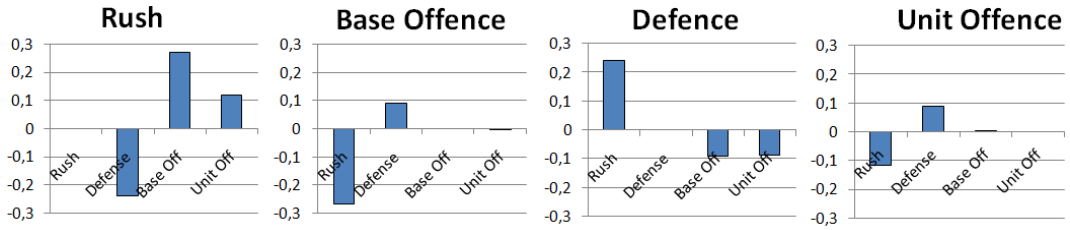


Figure 6: Fitnesses between each type of tactic.

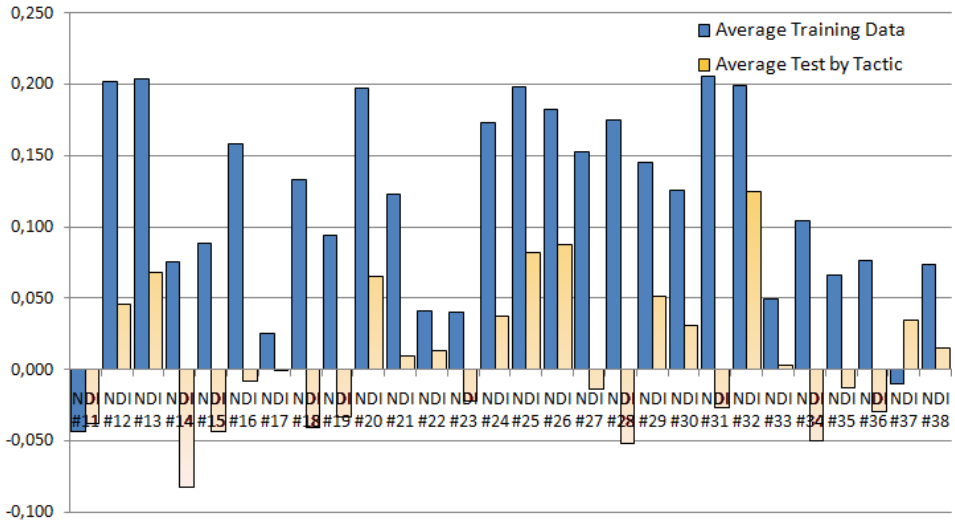


Figure 7: Average fitness of each non-dominated individual against the training set and the test set. The latter is weighed according to the prevalence of each tactic in the test set.

these opponents uses a different tactic, the AI must learn to outperform each of these tactics simultaneously, which potentially results in a generalised AI.

In the second experiment we decided to generate an AI against three of the different AIs from the first experiment, namely #0 (the Rush AI), #1 (a Defence tactic) and #3 (A Unit Offense tactic). The approach we used for this experiment is based on the SPEA2 algorithm [8]. It uses the concept of Pareto dominance to establish an ordering between different individuals. An individual is Pareto dominated if another individual's fitness values are at least equal to the first individuals fitness values, and at least one of these values is greater. Using this concept, non-dominated individuals can objectively be considered better than their dominated counterparts.

The SPEA2 approach differs from regular EAs in two important ways. Firstly, aside from the regular population, an archive containing non-dominated individuals is maintained. Each new population is now generated using the individuals from not just the population, but also from this archive. Secondly, two factors are used to determine the scalar fitness value of individuals, namely (1) the number of individuals which dominate this individual and (2) the location of this individual in the fitness space.

At the end of the second experiment (which we ran only once due to the time-intensive nature of the task), the learning algorithm had generated 28 non-dominated individuals. To determine the quality of these AIs, they were played against the test data, which consists of the previously generated AIs which were not used as training data (AIs #2 and #4 through #10). The resulting averages of the fitness values against both the training and the test sets are displayed in Figure 7. The test set averages are weighted according to the prevalence of their corresponding tactic within this set.

Unsurprisingly, the results against the test set are generally lower than those against the training set. However, in some cases the differences are exceedingly large. For example, the AI with the highest average fitness against the training set, Non-Dominated Individual (NDI) #31, has a negative average fitness against the test set. The fitness values for this individual imply that a single AI using a certain observed tactic, is not necessarily representative for other AIs using the same tactic. The test results in this figure also show a

clear favourite; NDI #32 outperforms the other AIs by a wide margin.

To obtain a more accurate measure of the generated AIs' performances, they were also made to play each other in a tournament. Again, NDI #32 proved to be the best-performing AI.

Analysing the behaviour of NDI #32 shows an interesting tactic. It does not simply rush toward enemy units or bases, but it is not strictly defensive either. For the first few turns, this AI does not move its units, until any attacking enemy units move into attack range. Then, the AI sends all of its units towards these enemy units. Because the enemy's units move at different rates and therefore do not arrive simultaneously, this results in a material advantage as well as the initiative of attack. This behaviour shows that the learning algorithm is able to generate an AI which not only outperforms multiple tactics, but does so using a new, somewhat more complicated tactic, instead of an improved version of the previously generated tactics.

8 Conclusions

The goal of this research was to generate automatically an effective TBS AI player. To accomplish this, our approach focussed on spatial reasoning, through the use of influence mapping. In this work, we have extended the influence mapping approach by implementing an improved layering algorithm, and by creating a learning algorithm which generates the weights, and therefore the behaviour, of the influence maps automatically. We have also shown that influence maps can be used to determine directly the behaviour of an AI player.

Moreover, in order to decompose the complex task of creating a TBS AI player, we proposed the ADAPTA architecture. This architecture makes it possible to concentrate AI design on a single subtask. This promotes the possibility to implement learning AIs.

In our experiments, we chose one subtask, namely an Extermination module which is aimed at combat. For this subtask we learned new AIs using an evolutionary algorithm. The results achieved showed that the ADAPTA AI is able to generate tactics that defeat all single opponents. Moreover, by learning against multiple opponents using different tactics simultaneously, an AI was created which was able to play at least equally well and outperform most of the previously generated AIs.

For future work, we intend to explore the capabilities and limitations of the auctioning mechanism in the strategic layer of the ADAPTA architecture, as this was not included in the present research. Furthermore, we intend to explore whether the successes achieved with learning of the Extermination module can be repeated for the other modules.

Acknowledgements

This research was funded by a grant from the Netherlands Organization for Scientific Research (NWO grant No 612.066.406).

References

[1] M. Bergsma. Adaptive spatial reasoning for turn-based strategy games. Masters thesis, Maastricht University, 2008.

[2] M. Buro and T. Furtak. RTS games as test-bed for real-time AI research. Workshop on Game AI, JCIS 2003, page 20, 2003.

[3] D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, 1989.

[4] M. Ponsen, H. Muñoz-Avial, P. Spronck and D.W. Aha. Automatically generating game tactics with evolutionary learning. AI Magazine, 27(3):75-84, 2006.

[5] P. Tozour. Influence mapping. Game Programming Gems, 2:287297, 2001.

[6] P. Tozour. Using a spatial database for runtime spatial analysis. AI Game Programming Wisdom, 2:381390, 2004.

[7] X. Yao. Evolving artificial neural networks. In Proceedings of the IEEE, volume 87, pages 14231447, 1999.

[8] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.

Attack Relations among Dynamic Coalitions

Guido Boella [a] Leendert van der Torre [b] Serena Villata [a]

[a] *University of Turin, Italy, email: {boella,villata}@di.unito.it*
[b] *University of Luxembourg, Luxembourg, email: leendert@vandertorre.com*

Abstract

In this paper we introduce a formal argumentation framework to reason about the evolution of coalitions. We extend Amgoud's preference-based argumentation framework for coalition formation to an argumentation framework with nested attack relations along the lines proposed by Modgil, and we define an instance of the argumentation framework for reasoning about coalitions based on a new dynamic extension of Conte and Sichman's static dependence networks. We show how this new approach covers a wider range of attacks than Amgoud's task based coalition formation, including attacks weakening the stability of the coalition by removing dependencies. Moreover, we show how it can be used not only for coalition formation, but also for coalitions which evolve due to the addition of dependencies.

1 Introduction

Dung's argumentation theory [11] may be seen as a formal framework for nonmonotonic logic and logic programming, and has been applied to many domains in which non-monotonic reasoning plays a role, such as decision making or coalition formation [4].

Amgoud [1] proposes to use Dung's argumentation theory and associated dialogue theories as a formal framework for coalition formation, and she illustrates this idea by formalizing a task based theory of coalition formation as in instance of Dung's argumentation theory. In this formalization, an argument is a set of agents together with a task, and an argument attacks another one if the two coalitions share an agent, or when they contain the same task. It is therefore based on strong assumptions, for example that an agent cannot be part of two coalitions at the same time. Since the attack relation is symmetric, also preferences are introduced to resolve conflicts.

In this paper, we are interested in a generalization of Amgoud's argumentation based coalition theory, covering a wider range of attacks, and a broader range of reasoning about coalitions. In particular, we would like to cover not only coalition formation but also the dynamics of coalitions, whose stability can be attacked and which can evolve by adding new dependencies among agents in the coalition. In this paper we address the following questions:

1. How to model preferences among coalitions? In particular, how can a global preference relation among coalitions be defined, or derived from the preference relations on coalitions of individual agents?
2. In which ways can a coalition attack another one, including attacks on the stability of coalitions? What kind of coalition formation theory should we use in the argumentation theory to cover a wider range of attacks among coalitions?
3. In which ways can a coalition evolve in time by adding new dependencies among agents in the coalition? How to cover not only coalition formation but also coalition evolution?

To represent preferences among coalitions in terms of attacks of attack relations, we use Modgil's argumentation theory [13]. Modgil has introduced an extension of Dung's framework in which arguments can be given why one argument is preferred to another one. In particular, he proposes an argumentation framework in which an argument can attack an attack relation, and illustrates how this can be used to represent arguments for preferences. In this paper we show how such attacks of attack relations can be modeled as an instance of Dung's theory.

To represent attacks on stability of coalitions and their evolution, we argue that instead of a task based coalition theory as used by Amgoud, a wider range of attacks can be defined using Sichman and Conte's dependence network theory. We extend their theory for conditional dependencies, in which agents can create or destroy dependencies by introducing or removing powers and goals of agents. Goals can be introduced if goals are conditional, or when the agent can create normative goals by creating obligations for the other agents.

The layout of this paper is as follows. In Section 2 we introduce Amgoud's argumentation theory for coalition formation, which is equivalent to Dung's abstract argumentation theory in which arguments are about coalitions, and we introduce our representation of Modgil's argumentation framework with nested attack relations as an instance of Dung's framework. In Section 3 we introduce a dynamic version of Conte and Sichman's dependence networks, following Caire et al [9] we introduce conditional dependencies, but we distinguish among adding powers and goals, and removing them. Finally, in Section 4 we illustrate the framework with examples concerning the deletion and addition of dependencies in coalitions using the argumentation theory. Related work and conclusions end the paper.

2 Arguing about coalitions

Argumentation is a reasoning model based on constructing arguments, identifying potential conflicts between arguments and determining acceptable arguments. Amgoud [1] proposes to use it to construct arguments to form coalitions, identify potential conflicts among coalitions, and determine the acceptable coalitions. Dung's framework [11] is based on a binary attack relation among arguments. In Dung's framework, an argument is an abstract entity whose role is determined only by its relation to other arguments. Its structure and its origin are not known. In this section, following Amgoud, we assume that each argument proposes to form a coalition, but we do not specify the structure of such coalitions yet. We represent the attacks among arguments by $\#$.

Definition 1 (Argumentation framework) *An argumentation framework is a pair $\langle \mathcal{A}, \# \rangle$, where $\mathcal{A}$ is a set (of arguments to form coalitions), and $\# \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation over $\mathcal{A}$ representing a notion of* attack *between arguments.*

The various semantics of an argumentation framework are all based on the notion of defense. A set of arguments $\mathcal{S}$ defends an argument a when for each attacker b of a, there is an argument in $\mathcal{S}$ that attacks b. A set of acceptable arguments is called an *extension*.

Definition 2 (Acceptable arguments)

- *$\mathcal{S} \subseteq A$ is attack free if and only if there are no arguments $a_1, a_2 \in \mathcal{S}$ such that a_1 attacks a_2.*
- *$\mathcal{S}$ defends a if and only if for all $a_1 \in \mathcal{A}$ such that a_1 attacks a, there is an alternative $a_2 \in \mathcal{S}$ such that a_2 attacks a_1.*
- *$\mathcal{S}$ is a preferred extension if and only if $\mathcal{S}$ is maximal w.r.t. set inclusion among the subsets of $\mathcal{A}$ that are attack free and that defend all their elements.*
- *$\mathcal{S}$ is a basic extension if and only if it is a least fixpoint of the function $F(\mathcal{S}) = \{a | a$ is defended by $\mathcal{S}\}$.*

The following example illustrates argumentation theory.

Example 1 *Let $AF = \langle \mathcal{A}, \# \rangle$ be an argumentation framework, where $\mathcal{A} = \{C_1, C_2, C_3\}$ is the set (of arguments or coalitions), and $\{C_1 \# C_2, C_2 \# C_3\}$ is the binary relation over $\mathcal{A}$ representing a notion of* attack *between arguments. Due to the so-called reinstatement principle of argumentation theory, the acceptable arguments are C_1 and C_3, for any kind of semantics. C_1 is accepted because it is not attacked by any other argument, and C_3 is accepted because its only attacker C_2 is attacked by an accepted argument.*

Amgoud [1] proposes to use preference-based argumentation theory for coalition formation, in which the attack relation is replaced by a binary relation $\mathcal{R}$, which she calls a defeat relation, together with a (partial) preordering on the coalitions. Each preference-based argumentation framework represents an argumentation framework, and the acceptable arguments of a preference-based argumentation framework are simply the acceptable arguments of the represented argumentation framework.

Definition 3 (Preference-based argumentation framework) *A preference-based argumentation framework is a tuple $\langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ where $\mathcal{A}$ is a set of arguments or coalitions, $\mathcal{R}$ is a binary defeat relation defined on $\mathcal{A} \times \mathcal{A}$ and $\succeq$ is a (total or partial) pre-order (preference relation) defined on $\mathcal{A} \times \mathcal{A}$. A preference-based argumentation framework $\langle \mathcal{A}, \mathcal{R}, \succ \rangle$ represents $\langle \mathcal{A}, \# \rangle$ if and only if $\forall a, b \in \mathcal{A}$, we have $a \# b$ if and only if $a \mathcal{R} b$ and it is not the case that $b \succ a$ (i.e., $b \succeq a$ without $a \succeq b$). The extensions of $\langle \mathcal{A}, \mathcal{R}, \succ \rangle$ are the extensions of the represented argumentation framework.*

The following example illustrates the preference based argumentation theory.

Example 2 (Continued) *Let $PAF = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ be a preference-based argumentation framework, where $\mathcal{A} = \{C_1, C_2, C_3\}$ is a set of arguments to form coalitions, $\{C_1 \mathcal{R} C_2, C_2 \mathcal{R} C_1, C_2 \mathcal{R} C_3, C_3 \mathcal{R} C_2\}$ a binary defeat relation defined on $\mathcal{A} \times \mathcal{A}$ and $\{C_1 \succ C_2, C_2 \succ C_3\}$ a total order (preference relation) defined on $\mathcal{A} \times \mathcal{A}$. PAF represents AF, so the acceptable arguments are again C_1 and C_3, for any kind of semantics.*

In general, preference-based argumentation frameworks are a useful and intuitive representation for argumentation frameworks, but for the application of coalition formation it is less clear where the preferences among coalitions come from. Moreover, when the defeat relation is symmetric, as in Amgoud's task based coalition theory, then it leads to a lack of expressive power, because some attack cycles can no longer be represented (see [12] for details).

Modgil [13] relates preferences to second-order attacks. Suppose that arguments a and b attack each other, and that argument a is preferred to argument b. Modgil observes that we can then say that the preference attacks the attack relation from b to a. The advantage of this perspective is that Modgil introduces also arguments which attack attack relations, which he uses to represent non-monotonic logics in which the priorities among the rules are represented in the formalism itself, rather than being given a priori (such as Brewka's theory [7], or Prakken and Sartor's theory [14]). Whereas Modgil presents his theory as an extension of Dung, such that he has to define new semantics for it, we define a version of second order attacks as an instance of Dung's theory. Each second order argumentation framework represents an argumentation framework, and the acceptable arguments of the second order argumentation framework are simply the acceptable arguments of the represented argumentation framework.

Definition 4 *A second order argumentation framework is a tuple $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, not, \mathcal{A}_\#, \# \rangle$, where $\mathcal{A}_\mathcal{C}$ is a set of coalition arguments, $\overline{\mathcal{A}}$ is a set of arguments such that $|\overline{\mathcal{A}}| = |\mathcal{A}_\mathcal{C}|$, not is a bijection from $\mathcal{A}$ to $|\overline{\mathcal{A}}|$, $\mathcal{A}_\#$ is a set of arguments that coalitions attack each other, and $\# \subseteq (\mathcal{A}_\mathcal{C} \times \overline{\mathcal{A}}) \cup (\overline{\mathcal{A}} \times \mathcal{A}_\#) \cup (\mathcal{A}_\# \times \mathcal{A}_\mathcal{C}) \cup (\mathcal{A}_\mathcal{C} \times \mathcal{A}_\#)$ is a binary relation on the set of arguments such that for $a \in \mathcal{A}_\mathcal{C}$ and $b \in \overline{\mathcal{A}}$ we have $a \# b$ if and only if $b = not(a)$, and for each $a \in \mathcal{A}_\#$, there is precisely one $b \in \overline{\mathcal{A}}$ such that $b \# a$ and precisely one $c \in \mathcal{A}_\mathcal{C}$ such that $a \# c$. A second order argumentation framework $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, \mathcal{A}_\#, \# \rangle$ represents $\langle \mathcal{A}, \# \rangle$ if and only if $\mathcal{A} = \mathcal{A}_\mathcal{C} \cup \overline{\mathcal{A}} \cup \mathcal{A}_\#$. The extensions of $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, \mathcal{A}_\#, \# \rangle$ are the extensions of the represented argumentation framework.*

The following example illustrates the second order argumentation theory. The main feature of $not(C)$ arguments is just to ensure that if an argument is not accepted, then it cannot attack other arguments. The argument C_0 is a dummy argument to represent the preferences, here used to map the second order framework to the preference-based one. This example is showed in Figure 1.

Example 3 (Continued) *Let $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, not, \mathcal{A}_\#, \# \rangle$ be a second order argumentation framework, where $\mathcal{A}_\mathcal{C} = \{C_1, C_2, C_3, C_0\}$ is a set of coalition arguments, $\overline{\mathcal{A}} = \{C_1', C_2', C_3', C_0'\}$, not is the bijection $not(C_i) = C_i'$, $\mathcal{A}_\# = \{C_{1,2}, C_{2,1}, C_{2,3}, C_{3,2}\}$ is a set of arguments that coalitions attack each other, and*

$$\{C_1 \# C_1', C_2 \# C_2', C_3 \# C_3', C_0 \# C_0', C_1' \# C_{1,2}, C_2' \# C_{2,1}, C_2' \# C_{2,3}, C_3' \# C_{3,2},$$
$$C_{1,2} \# C_2, C_{2,1} \# C_1, C_{2,3} \# C_3, C_{3,2} \# C_2, C_0 \# C_{2,1}, C_0 \# C_{3,2}\}$$

is a binary relation on the set of arguments. For the nested attack relations, we also write $C_0 \#(C_2 \# C_1)$ and $C_0 \#(C_3 \# C_2)$. The acceptable arguments are C_1 and C_3, together with C_0, C_2', $C_{1,2}$, $C_{2,3}$, for any kind of semantics.

We use the same example in Section 4 using coalitions defined by dynamic dependence networks.

We can visualize second order argumentation frameworks by not visualizing $\overline{\mathcal{A}}$ or $\mathcal{A}_\#$, and visualizing an indirect attack from an element of $\mathcal{A}_\mathcal{C}$ to $\mathcal{A}_\mathcal{C}$ via an element of $\mathcal{A}_\#$ as an arrow, and an attack of an element

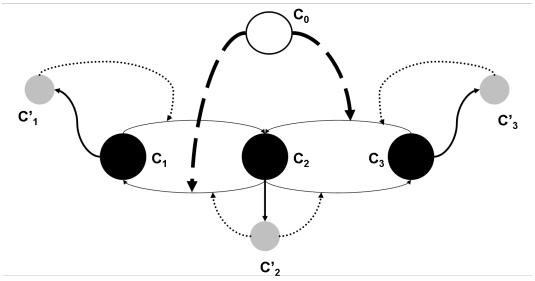


Figure 1: Graphical representation of Example 3.

of $\mathcal{A}_\mathcal{C}$ to an element of $\mathcal{A}_\#$ as an attack on an attack relation, see [13] for examples of such a visualization. Example 3 shows that arguments that attack attack relations do that directly. However, Barringer, Gabbay and Woods [3] argue that such attack relations can themselves also be attacked, leading to a notion of higher order attacks.

Definition 5 *A higher order argumentation framework is a tuple $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, not, \mathcal{A}_\#, \# \rangle$, where $\mathcal{A}_\mathcal{C}$ is a set of coalition arguments, $\overline{\mathcal{A}}$ is a set of arguments such that $|\overline{\mathcal{A}}| = |\mathcal{A}_\mathcal{C}|$, not is a bijection from $\mathcal{A}$ to $|\overline{\mathcal{A}}|$, $\mathcal{A}_\#$ is a set of arguments that coalitions attacks attack each other, and $\# \subseteq (\mathcal{A}_\mathcal{C} \times \overline{\mathcal{A}}) \cup (\overline{\mathcal{A}} \times \mathcal{A}_\#) \cup (\mathcal{A}_\# \times \mathcal{A}_\mathcal{C}) \cup (\mathcal{A}_\# \times \mathcal{A}_\#)$ is a binary relation on the set of arguments such that for $a \in \mathcal{A}_\mathcal{C}$ and $b \in \overline{\mathcal{A}}$ we have $a \# b$ if and only if $b = not(a)$, and for each $a \in \mathcal{A}_\#$, there is precisely one $b \in \overline{\mathcal{A}}$ such that $b \# a$ and precisely one $c \in \mathcal{A}_\mathcal{C} \cup \mathcal{A}_\#$ such that $a \# c$. A higher order argumentation framework $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, \mathcal{A}_\#, \# \rangle$ represents $\langle \mathcal{A}, \# \rangle$ if and only if $\mathcal{A} = \mathcal{A}_\mathcal{C} \cup \overline{\mathcal{A}} \cup \mathcal{A}_\#$. The extensions of $\langle \mathcal{A}_\mathcal{C}, \overline{\mathcal{A}}, \mathcal{A}_\#, \# \rangle$ are the extensions of the represented argumentation framework.*

3 Coalition formation

Coalitions can be defined in so-called dependence networks, based on the idea that to be part of a coalition, every agent has to contribute something, and has to get something out of it. Dependence networks are a kind of social network introduced by Sichman and Conte, representing how each agent depends on other agents to achieve the goals he cannot achieve himself. They are used to specify early requirements in the Tropos agent methodology [6], and to model and reason interactions among agents in multiagent systems. Roughly, a coalition can be formed when there is a cycle of dependencies (the definition of coalitions is more complicated due to the fact that an agent can depend on a set of agents, see below).

Dynamic dependence networks have been introduced by Caire et al. [9], in which a dependency between agents can depend on the interaction of other agents. Here we distinguish "negative" dynamic dependencies where a dependency exists unless it is removed by a set of agents, due to removal of a goal or ability of an agent, and "positive" dynamic dependencies where a dependency may be added due to the power of a third set of agents. As explained in the following section, these two dynamic dependencies can be used to reason about the evolution of coalitions.

Definition 6 (Dynamic Dependence Networks) *A dynamic dependence network is a tuple $\langle A, G,$ dyndep$^-$, dyndep$^+ \rangle$ where:*

- *A is a set of agents and G is a set of goals.*
- *dyndep$^-$: $A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of a agent and two sets of agents all the sets of goals in which the first depends on the second, unless the third deletes the dependency. The static dependencies are defined by dep(a, B) = dyndep$^-(a, B, \emptyset)$.*
- *dyndep$^+$: $A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of a agent and two sets of agents all the sets of goals on which the first depends on the second, if the third creates the dependency.*

A coalition can be represented by a set of dependencies, represented by $C(a, B, G)$ where a is an agent, B is a set of agents and G is a set of goals. Intuitively, the coalition agrees that for each $C(a, B, G)$ part of the coalition, the set of agents B will see to the goal G of agent a. Otherwise, the set of agents B may be removed from the coalition or be sanctioned.

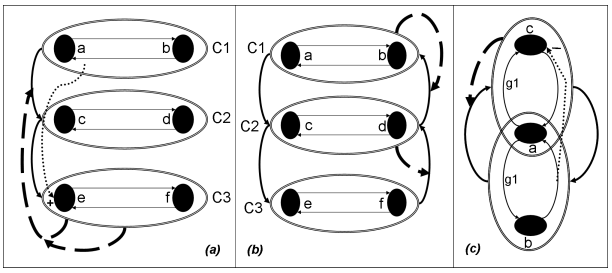


Figure 2: (a) - (b) - Three coalitions attacking each other, (c) - Vulnerable coalitions

Definition 7 *Let A be a set of agents and G be a set of goals. A coalition function is a partial function $C : A \times 2^A \times 2^G$ such that$\{a \mid C(a, B, G)\} = \{b \mid b \in B, C(a, B, G)\}$, the set of agents profiting from the coalition is the set of agents contributing to it. Let $\langle A, G, dyndep^-, dyndep^+ \rangle$ be a dynamic dependence network, and dep the associated static dependencies.*

1. *A coalition function C is a coalition if $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in dep(a, B)$. These coalitions which cannot be destroyed by addition or deletion of dependencies by agents in other coalitions.*
2. *A coalition function C is a vulnerable coalition if it is not a coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in \cup_D dyndep^-(a, B, D)$. Coalitions which do not need new goals or abilities, but whose stability can be destroyed by removing dependencies.*
3. *A coalition function C is a potential coalition if it is not a coalition or a vulnerable coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in \cup_D (dyndep^-(a, B, D) \cup G' \in dyndep^+(a, B, D))$ Coalitions which could be created or which could evolve if new abilities or goals would be created by agents of other coalitions on which they dynamically depend.*

In this paper we do not consider further refinements of the notion of coalition as in [5], but focus on the use of argumentation theory to reason about coalitions.

The basic attack relations between coalitions are due to the fact that they are based on the same goals. This is analogous to the conflicts between coalitions in Amgoud's coalition theory where two coalitions are based on the same tasks.

Definition 8 *Coalition C_1 attacks coalition C_2 if and only if there exists a, a', B, B', G, G', such that $C_1(a, B, G)$, $C_2(a', B', G')$ and $G \cap G' \neq \emptyset$.*

We illustrate the conflict among coalitions with an example:

Example 4 *Assume we have three agents, a, b, c and the dependencies (we write $C(a, b, g_1)$ for $C(a, \{b\}, \{g_1\})$ and $dep(a, b, g_1)$ for $dep(a, \{b\}, \{g_1\})$): $dep(a, b, g_1), dep(a, c, g_1), dep(b, a, g_2), dep(c, a, g_3)$. So there are two coalitions: $C_1 = \{(a, b, g_1), (b, a, g_2)\}, C_2 = \{(a, c, g_1), (c, a, g_3)\}$. They will not create both since one is enough for agent a to have someone look after his goal g_1: $C_1 \# C_2$ and $C_2 \# C_1$.*

We now go beyond Amgoud's approach by defining the second order attacks. The simplest kind of attack on an attack relation is to remove or add one of the dependencies of the attacker.

Definition 9 *Coalition C attacks the attack from coalition C_1 on coalition C_2 if and only if there exists a set of agents $D \subseteq \{a \mid \exists E, HC(a, E, H)\}$ such that $\exists a, B, G' C_1(a, B, G')$ and $G \in dyndep^{\{+,-\}}(a, B, D)$.*

On the one hand, this definition reflects the idea that the stability of a vulnerable coalition $C1$ can be endangered by agents of another coalition C if they decide to remove a dependency of $C1$ due to the dynamic dependency $dyndep^-(a, B, D)$. On the other hand that a potential coalition $C1$ can never materialize or evolve if the agents of $C1$ do not create the dependency denoted by $dyndep^+(a, B, D)$.

The effect of making a vulnerable coalition unstable or of leaving a potential coalition immaterial is represented by the fact that all the attack relations which stem from it are attacked by the coalition C on which $C1$ dynamically depend. The next section will discuss these two possibilities thanks to some examples.

4 Examples

In this section we illustrate by means of examples how to reason about coalitions in the argumentation framework. The first example shows three coalitions which attack each other since they share some goals. Attacks on attacks relations allow to define asymmetric attacks.

Example 5 *Assume we have six agents, a, b, c, d, e, f and the following dependencies:*

$dep(a, b, g_1)$, $dep(b, a, g_2)$, $dep(c, d, g_1)$, $dep(d, c, g_3)$, $dep(e, f, g_4)$, $dep(f, e, g_3)$, $dep(a, d, g_1)$, $dep(c, b, g_1)$, $dep(d, e, g_3)$, $dep(f, c, g_3)$.

The possible coalitions are C_1, C_2 and C_3 where $C_1 = \{(a, b, g_1), (b, a, g_2)\}$, $C_2 = \{(c, d, g_1), (d, c, g_3)\}$, $C_3 = \{(e, f, g_4), (f, e, g_3)\}$.

Note that some of the dependencies remain outside all coalitions (e.g., $dep(a, d, g_1)$, $dep(c, b, g_1)$). Thus, $C_1 \# C_2$, $C_2 \# C_1$, $C_2 \# C_3$ and $C_3 \# C_2$ due to the fact that they share goals g_1 and g_3 respectively. Note that these attacks are reciprocal.

The coalitions attack each other since, for example, agents b and d on which respectively a and c depend for g_1 would not make their part hoping that the other one will do that, so to have a free ride and get respectively g_2 achieved by a and g_3 by c.

To model the fact that C_1 is more important than C_2 and C_2 of C_3 we add an attack on the attack relation: $C_1 \# (C_2 \# C_1)$ and $C_2 \# (C_3 \# C_2)$. Thus the only possible extension is $\{C_1, C_3\}$.

We depict this situation in Figure 2 - (b): normal arrows connecting the agents represent the dependencies among these agents (they can be labeled with the goal on which the dependence is based), coalitions are represented by the ovals containing the agents of the coalition, bold arrows indicate the attack relations among the coalitions and the attack relation on attack relations is depicted as bold dashed arrows pointing on other arrows. Using the terminology of argumentation theory, coalition C_1 attacks coalition C_2 but not vice versa, and C_2 attacks C_3 but not vice versa. Therefore C_1 becomes an acceptable coalition since it is not attacked, and C_3 is attacked only by a coalition which will not be accepted, and therefore it is reinstated as an acceptable coalition. This can be explained as follows. Since an agent of coalition C_1 has the power to destroy the coalition C_2 by removing a dependency, it will not be realized. The agents involved will prefer coalition C_1, which cannot be attacked by removing a dependency, and which is therefore more stable. Note that for this reasoning, the coalitions do not have to be constructed. It is assumed that all the agents know the dependencies, and they therefore realize that coalition C_2 is not viable. For coalition C_3, the situation is more complicated. Though it can be attacked by coalition C_2, the agents realize following the argumentation above that C_2 will never be realized, and therefore they realize that the agents in coalition C_2 who could attack the coalition, will not do so, because they have nothing to win by attacking the coalition. So despite the fact that it is a vulnerable coalition, it will be accepted by the agents involved. This follows by hypothetical reasoning using the dynamic dependence network, formalized in the argumentation theory based on the reinstatement principle.

The following example illustrates how the deletion of a dependency can be used to attack an attack relation (see Figure 2 - (c), dotted arrows characterized by a label + or - links a dependency with an agent and it indicates that the dependency can be added (+) or deleted (-)).

Example 6 (Continues Example 4) *Now, assume agent c can destroy the dependency $dep(b, a, g_2)$, i.e., we substitute it with dyndep$^-(b, a, c, g_2)$, for example by removing the power of a to see to goal g_2, or by removing the goal g_2 of agent b. This deletion allows agent c to ensure himself the dependence on himself of agent a on goal g_1. This deletion sets a preference relation of the coalition C_2, represented here with the attack of coalition C_2 to the attack relation of coalition C_1 to coalition C_2. In this case, the coalition $C_2 = \{(a, c, g_1), (c, a, g_3)\}$ will become the only possible extension, since $C_2 \# (C_1 \# C_2)$ by Definition 9.*

Whereas the previous example illustrates the role of dynamic dependencies which can delete existing dependencies, in the next example we consider the role of adding dependencies (see Figure 2 - (a)).

Example 7 (Continues Example 5) *Assume instead that $dep(b, a, g_2)$ is not present since the beginning and it happens that agent e of C_3 has the power to create it: i.e., it is substituted by dyndep$^+(b, a, e, g_2)$. Thus, C_3 attacks the attack relation between C_1 and C_1, $C_3 \# (C_1 \# C_2)$ by Definition 9: if coalition C_1 remains potential, then it cannot attack any other coalition. Thus, the only extension is $\{C_2\}$.*

However this dependence network does not capture the fact that agent e should be able to understand that it is better off in a situation where he actually creates the dependency $dep(b, a, g_2)$. We can represent

this situation by adding an attack relation between coalition C_3 which e belongs to and the attack relation between introduced by Definition 9: $C_3\#(C_3\#(C_1\#C_2))$. Note that this requires an higher order argumentation framework, like the one of Definition 5, where arguments can attack attack relations against other attack relations. This means that agent e exercises his option to add the dependency $dep(b, a, g_2)$ to C_1. The extension in this case would be $\{C_1, C_3\}$.

5 Related work

Although there were many approaches defining coalition formation, two represents different perspectives: the model of Shehory and Kraus [15] and the one of Sichman [16]. The approach of Shehory and Kraus [15] is based on the assumption that autonomous agents in the multiagent environments may need to cooperate in order to fulfill tasks. They present algorithms that enable the agents to form groups and assign a task to each group, calling these groups coalitions. The paper presents coalition formation algorithms which are appropriate for Distributed Problem Solving cases where agents cooperate in order to increase the overall outcome of the system and are not concerned with their personal payoffs as they are in MAS.

Sichman [16], instead, introduces a different point of view. He presents coalition formation using a dependence-based approach based on the notion of social dependence introduced by Castelfranchi [10]. Concerning coalition formation, this model introduces the notion of dependence situation, which allows an agent to evaluate the susceptibility of other agents to adopt his goals, since agents not automatically adopt the goals of each other. In this dependence-based model, coalitions can be modeled using dependence networks developed by Sichman and Conte [17] where an agent is described by a set of prioritized goals and a global dependence relation that explicates how an agent depends on other agents for fulfilling its goals. A definition of coalitions inspired by dependence networks is given by Boella et al. [5].

Once represented the internal structure of coalitions, one could study which kind of relations there are among potential coalitions at an higher level of detail disregarding which are the causes for incompatibility. The application of argumentation frameworks to coalition formation has been discussed by Amgoud [1] and by Bulling et al. [8]. Amgoud [1] provides a unified and general formal framework for generating the coalitions structures. The coalition formation problem is represented by Amgoud by means of four steps: constructing the coalitions, defining the defeasibility and preference relations between these coalitions, defining the acceptable coalitions and concluding. In contrast with our approach, a coalition is viewed as an abstract entity whose role is only determined by its relation to other coalitions and its structure is not known. Unlike Amgoud's work [1], we do not provide this paper with a proof theory since it is derivable from the argumentation theory's literature.

Bulling et al. [8], instead, combine the argumentation framework and ATL. They provide a formal extension of ATL in which the actual computation of the coalition is modeled in terms of argumentation semantics. A difference regarding the Amgoud's paper, is the intuition, in accordance with ATL, where larger coalitions are more powerful than smaller ones. Bulling's approach is a generalization of Dung's argumentation framework, extended with a preference relation. The basic notion is that of a coalitional framework containing a set of elements (agents or coalitions), an attack relation for modeling conflicts, and a preference relation between these elements to describe favorite agents/coalitions. The notion of coalitional framework is based on the notion of framework for generating coalition structures presented in [1].

6 Conclusions

We generalize Amgoud's argumentation-based coalition theory, covering a wider range of attacks, and a broader range of reasoning about coalitions.

To represent preferences among coalitions in terms of attacks of attack relations, we use Modgil's argumentation theory [13]. Instead of resolving conflicts among coalitions by preferences among coalitions, we can resolve conflicts by arguments that resolve them. Whereas preferences among coalitions seem a derived notion, we show that attacks of attacks among coalitions can be given a natural interpretation. Instead of a task-based coalition theory as in [1], a wider range of attacks can be defined using Sichman and Contes dependence network theory. The basic attack relations between coalitions are due to the fact that they are based on the same goals. Moreover, we go beyond Amgouds approach by defining a second order attack: the simplest kind of attack on an attack relation is to remove or add one of the dependencies of the attacker. In this way, we cover not only coalition formation but also the evolution of coalitions, whose stability can

be attacked by other coalitions. To represent attacks on stability of coalitions and their evolution we use an extension of Sichman and Conte's dependence network theory with dynamic dependencies. A coalition can be attacked by removing a dependency from the coalition, or by introducing new dependencies which create new coalitions.

Concerning future work, we would like to model the argumentation theory in the more general framework introduced recently by Baroni and Giacomin [2]. In their approach, an argumentation framework is interpreted as a set of arguments of a reasoner at a moment in time. Consequently, it can be used to model a dialogue among agents as a sequence or tree of Dung's argumentation frameworks. Deleting a dependency or restraining from creating a dependency can be used to defend a coalition from a direct or indirect attack. Conversely, restraining from deleting a dependency and creating one can be used to avoid attacking a coalition which counter attacks other coalitions. Since dynamic dependencies are mapped on arguments which attack attack relations, this mechanism should be made more flexible to adapt to the context. Moreover, we would like to explain the dynamics of dependence networks using a normative system: in this way it would be possible to sanction agents who do not fulfill their role in a coalition. Further kind of attacks among coalitions may be defined, for example by introducing incompatibilities among dependence relations.

References

[1] L. Amgoud, An Argumentation-Based Model for Reasoning About Coalition Structures, *Proceedings of ArgMAS'05*, pp. 217–228, (2005).

[2] P. Baroni and M. Giacomin, On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.* 171(10-15): 675-700, (2007).

[3] H. Barringer, D. M. Gabbay and J. Woods, Temporal Dynamics of Support and Attack Networks: From Argumentation to Zoology. *Mechanizing Mathematical Reasoning*, pp. 59–98, (2005).

[4] T. J. M. Bench-Capon and P. E. Dunne, Argumentation in artificial intelligence. *Artif. Intell.* 171(10-15): 619-641, (2007).

[5] G. Boella, L. Sauro and L. van der Torre, Strengthening Admissible Coalitions, *Proceedings of ECAI'06*, (2006).

[6] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos, Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Systems Journal*, 8, pp. 203–236, (2004).

[7] G. Brewka, Reasoning about Priorities in Default Logic, *Proceedings of AAAI*, pp. 940-945, (1994).

[8] N. Bulling, C. I. Chesnevar and J. Dix, Modelling Coalitions: ATL + Argumentation, *Proceedings of AAMAS'08*, pp. 681–688, (2008).

[9] P. Caire, S. Villata, L. van der Torre and G. Boella, Conviviality Masks in Multiagent Systems, *Proceedings of AAMAS'08*, pp. 1265–1268, (2008).

[10] C. Castelfranchi, The micro-macro constitution of power, *Protosociology*, 18, pp. 208–269, (2003).

[11] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.*, 77(2), pp. 321–357, (1995).

[12] S. Kaci, L. van der Torre and E. Weydert, On the Acceptability of Incompatible Arguments. *Proceedings of ECSQARU'07*, pp. 247–258, (2007).

[13] S. Modgil, An Abstract Theory of Argumentation That Accommodates Defeasible Reasoning About Preferences. *Proceedings of ECSQARU'07*, pp. 648-659, (2007).

[14] H. Prakken and G. Sartor, Argument-Based Extended Logic Programming with Defeasible Priorities. *Journal of Applied Non-Classical Logics*, 7(1), (1997).

[15] O. Shehory and S. Kraus, Methods for task allocation via agent coalition formation, *Artif. Intell.*, 101, pp. 165–200, (1998).

[16] J. S. Sichman, DEPINT: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario, *Journal of Artificial Societies and Social Simulation*,1(2), (1998).

[17] J. S. Sichman and R. Conte, Multi-agent dependence by dependence graphs, *Proceedings of AAMAS'02*, pp. 483–490, (2002).

Loopy Propagation: the Posterior Error at Convergence Nodes

Janneke H. Bolt Linda C. van der Gaag

Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089 3508 TB Utrecht, The Netherlands

Abstract

The nodes of a loop in a Bayesian network can be distinguished in nodes with just one incoming arc on the loop and nodes with two or more incoming arcs on the loop. The former will be called inner nodes and the latter will be called convergence nodes. During loopy propagation, an algorithm for approximate inference with Bayesian networks, cycling errors may arise at the inner nodes and convergence errors may arise at the convergence nodes. In a network in its prior state, only convergence errors may arise. In previous research, we studied this prior error in the probabilities computed for the convergence nodes. In a network in its posterior state, the convergence error may change in size and moreover a cycling error may enter the approximations. In this paper we study the posterior error found in the probabilities computed for a convergence node in a binary network with just a simple loop.

1 Introduction

A Bayesian network is a concise representation of a joint probability distribution over a set of stochastic variables. It consists of a directed acyclic graph and a set of conditional probability distributions [7]. From a Bayesian network, in theory, any probability of the represented distribution can be infered. Inference, however, is NP-hard in general [4] and may be infeasible for large, densely connected networks. For networks for which exact inference is infeasible, approximate algorithms have been designed. A well-known algorithm for approximate reasoning with a Bayesian network is the *loopy-propagation algorithm*. For this algorithm good approximation performance has been reported, for example, [6]. The basic idea of this algorithm is to apply Pearl's propagation algorithm, an algorithm for exact inference with singly connected networks, to a network regardless of its topological structure. Many researchers have addressed the performance of the loopy-propagation algorithm analytically, for example, [8, 9]. Most of those studies, however, concern an equivalent algorithm on undirected networks. The investigation of the equivalent algorithm is motivated by the relatively easier analysis and justified by the observation that any Bayesian network can be converted into an undirected network.

In previous research we studied loopy propagation in Bayesian networks and found that two types of error may arise in the computed probabilities: cycling errors and convergence errors [1]. In the directed graph of a Bayesian network loop nodes can be distinguished in nodes with just one incoming arc on the loop and loop nodes with two or more incoming arcs on the loop. We called the former nodes inner nodes and the latter convergence nodes. A cycling error arises at the inner nodes of a loop and co-occurs with the cycling of messages in a loop during propagation. This type of error will only be found in a network in its posterior state. A convergence error arises at the convergence nodes of a loop and results from the combination of messages from the parents of a convergence node as if these parents were independent. This type of error may arise in a network in its prior as well as in a posterior state. In [1] we identified the factors that govern the size of the convergence error in the probabilities computed for the convergence node in a binary network with just a simple loop in its prior state and in [3] we conjectured a general expression for the prior convergence error.

When observations have been entered into a network, the error in the approximations for the convergence node may change in size. First of all, the convergence error itself may change and moreover a cycling error

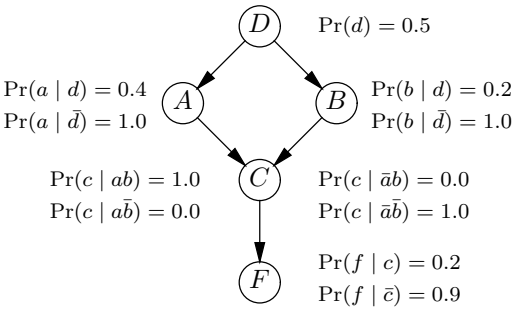


Figure 1: An example Bayesian network with a convergence node C having the dependent parents A and B and the child F.

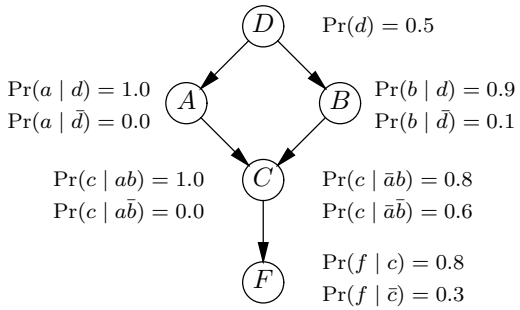


Figure 2: An example Bayesian network with a convergence node C having the dependent parents A and B and the child F.

may enter the approximation through a cycling error in the messages from its parent nodes. In this paper we investigate, for binary networks with just a simple loop, the posterior error found in the probabilities computed for the convergence node. We thereby first study the posterior convergence error and then describe the additional effect of the cycling error. More details about the research described in this paper can be found in [2].

The paper is organised as follows. Section 2 provides some preliminaries on Bayesian networks; it further introduces the loopy-propagation algorithm and reviews the two types of error that can arise during loopy propagation. Section 3 summarises the results from our previous study of the prior convergence error in [1]. In Section 4, the posterior error at the convergence nodes is discussed. The paper ends with our conclusions in Section 5.

2 Preliminaries

2.1 Bayesian Networks

A *Bayesian network* is a graphical model of a joint probability distribution $\Pr$ over a set of stochastic variables. In this paper all variables are assumed to be binary, taking one of the values *true* and *false*. Single variables are denoted by upper-case letters (A), possibly supplemented with a superscript (A^i). The value *true* of A will be written as a and the value *false* as $\bar{a}$; a_i is used to denote any value assignment to A, that is $a_i \in \{a, \bar{a}\}$. Sets of variables are denoted by bold-face upper-case letters ($\mathbf{A}$) and a joint value assignment to such a set is indicated by a bold-face lower-case letter ($\mathbf{a}$). The upper-case letter is also used to indicate the whole range of possible value assignments. In a Bayesian network, each variable of the modelled distribution is represented by a node in a directed acyclic graph[1]. Independencies between the variables are, as far as possible, captured by the digraph's set of arcs according to the d-separation criterion [7]. Moreover, for each variable A, conditional probability distributions $\Pr(A \mid \pi(A))$ are specified, where $\pi(A)$ denotes the joint value assignments to the set of parents of A in the digraph. A network is *singly connected* if there is at most one trail between any two variables in its digraph. Otherwise, it is *multiply connected*. A multiply connected network includes one or more loops, that is, one or more cycles in its underlying undirected graph; we will call a loop simple, if none of its arcs is shared by another loop. A node that has just one incoming arc on the loop will be called an *inner node*; a loop node with two or more incoming arcs on the loop will be called a *convergence node*. Figure 1 depicts a small example Bayesian network. This network is multiply connected. The trail $A \leftarrow D \rightarrow B \rightarrow C \leftarrow A$ constitutes a simple loop; node C is the only convergence node of this loop.

2.2 Loopy Propagation

Pearl's propagation algorithm [7] is designed for exact probabilistic inference with singly connected Bayesian networks. The term loopy propagation is used throughout the literature to refer to the application of this algorithm to networks with loops. In the algorithm, each node $X \in \mathbf{V}$ has a limited set of rules with which it can compute its marginal probability distribution $\Pr(X \mid \mathbf{o})$ from messages it receives from its neighbours.

[1] From now on, the terms node and variable will be used interchangeably.

The messages that a node receives from its parents are called *causal messages* and the messages that a node receives from its children are called *diagnostic messages*. A node uses the causal messages it receives to compute its *compound causal parameter* and it uses the diagnostic messages it receives to compute its *compound diagnostic parameter*. A node combines its compound parameters to obtain its marginal distribution given the observations $\mathbf{o}$. The rules of the algorithm are applied in parallel by the various nodes at each time step. The rule used by node X for establishing the probability distribution $\Pr(X \mid \mathbf{o})$ at time t is

$$\Pr^t(X \mid \mathbf{o}) = \alpha \cdot \lambda^t(X) \cdot \pi^t(X)$$

where the compound diagnostic parameter $\lambda^t(X)$ is computed from the diagnostic messages $\lambda^t_{Y^j}(X)$ it receives from each of its children Y^j

$$\lambda^t(X) = \prod_j \lambda^t_{Y^j}(X)$$

and the compound causal parameter $\pi^t(X)$ is computed from the causal messages $\pi^t_X(U^i)$ it receives from each of its parents U^i

$$\pi^t(X) = \sum_{\mathbf{U}} \Pr(X \mid \mathbf{U}) \cdot \prod_i \pi^t_X(U^i)$$

where $\mathbf{U}$ denotes the set of all parents of node X. The rule for computing the diagnostic messages to be sent to its parent U^i is

$$\lambda^{t+1}_X(U^i) = \alpha \cdot \sum_X \lambda^t(X) \cdot \sum_{\mathbf{U}/\{U^i\}} \Pr(X \mid \mathbf{U}) \cdot \prod_{k \neq i} \pi^t_X(U^k)$$

and the rule for computing the causal messages to be sent to its child Y^j is

$$\pi^{t+1}_{Y^j}(X) = \alpha \cdot \pi^t(X) \cdot \prod_{k \neq j} \lambda^t_{Y^k}(X)$$

In the above computation rules, α denotes a normalisation constant. In a singly connected network, after a limited number of time steps an equilibrium state is reached in which $\Pr(X \mid \mathbf{o}) = \alpha \cdot \lambda(X) \cdot \pi(X)$. When applied to a multiply connected network, the algorithm will often reach an equilibrium state as well, but the computed probabilities may now include errors. In [1], we distinguished between cycling errors and convergence errors. A *cycling error* arises at the inner nodes of a network and co-occurs with a cycling of messages in the loop. Such an error will only arise in a network in its posterior state. As an example, consider the network from Figure 1. After an observation for node C has been entered, the diagnostic message that C sends to node A will consist of information about its own conditional probabilities combined the probability distribution of node B. Node A passes this information on to node D which in turn sends it to B. Node B misinterprets its own information for new and includes it into its probability distribution. Node B then sends a different message to node C and the updating process is repeated. A *convergence error* arises at the convergence nodes of a network and may arise in a network in its prior state as well as in a posterior state. A convergence error may arise because a convergence node combines the messages from its parents as if these parents were independent while, in fact, they may be dependent. As an example consider again the network from figure 1. Node C combines the messages from A and B as if nodes A and B were independent, while in fact, they are dependent along the trail $A \leftarrow D \rightarrow B$. In a network in its prior state, only a convergence error may be found at the convergence nodes; in a network in a posterior state, however, the messages that a convergence node receives from its parents may include a cycling error. In the probabilities computed for the convergence node in a posterior state, therefore, also a cycling error may be introduced.

In the sequel we will use a tilde to distinguish the approximate probabilities and parameters that result upon loopy propagation from their exact counterparts.

3 The Prior Error at Convergence Nodes

In [1], we studied the prior error that arises upon loopy propagation in a convergence node of a simple loop in a binary network. Below the obtained results are briefly reviewed. Consider the example network from Figure 1. The nodes A and B are dependent along the trail $A \leftarrow D \rightarrow B$. This, however, is not taken into

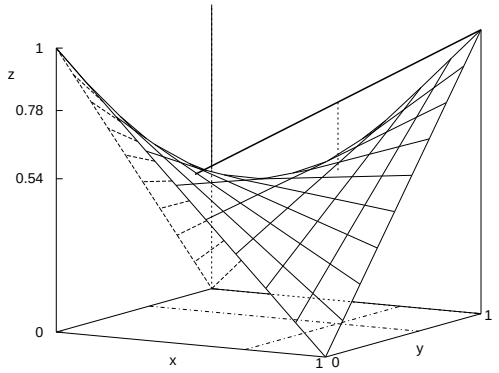


Figure 3: The line segment capturing $\Pr(c)$ and the surface capturing $\widetilde{\Pr}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ for the network from Figure 1.

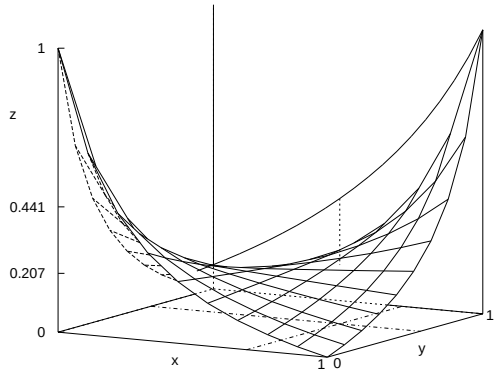


Figure 4: The line segment capturing $\Pr(c \mid f)$ and the surface capturing $\widetilde{\Pr}_{conv}(c \mid f)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ for the network from Figure 1.

account by Pearl's propagation algorithm. As a result, the difference between the exact probability $\Pr(c)$, and the approximate probability $\widetilde{\Pr}(c)$ calculated by Pearl's algorithm equals

$$\begin{aligned}\Pr(c) - \widetilde{\Pr}(c) \;=\; & \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) \cdot \\ & \big(\Pr(a \mid d) - \Pr(a \mid \bar{d})\big) \cdot \\ & \big(\Pr(b \mid d) - \Pr(b \mid \bar{d})\big) \cdot \\ & \big(\Pr(d) - \Pr(d)^2\big)\end{aligned}$$

In the sequel, v is used to denote the prior convergence error $\Pr(c) - \widetilde{\Pr}(c)$. The absolute value of the v ranges between 0 and 0.5.

The prior convergence error, is illustrated in Figure 3; for the construction of the figure we used the network from Figure 1. The line segment captures the exact probability $z = \Pr(c)$ as a function of $\Pr(d)$, given the conditional probabilities as specified for the nodes A, B and C of the example network. Note that each specific $\Pr(d)$ corresponds with a unique combination of $x = \pi_C(a)$ and $y = \pi_C(b)$. The depicted surface captures $z = \widetilde{\Pr}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$, given the conditional probabilities as specified for C of the example network. The approximate probability $\widetilde{\Pr}(c)$ is computed from the exact messages $\pi_C(a)$ and $\pi_C(b)$, and therefore, 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 the $\Pr(d) = 0.5$ from the network, the difference between $\Pr(c)$ and $\widetilde{\Pr}(c)$ is indicated by the vertical dotted line segment and equals $0.78 - 0.54 = 0.24$.

4 The Posterior Error at Convergence Nodes

In this section the posterior error found in the probabilities computed for the convergence node in a network with just a simple loop is investigated. In Section 4.1 we thereby abstract from the cycling error by considering the error which would be found given that the convergence node receives the exact causal messages from its parent nodes. Thereafter, in Section 4.2, the effect of the cycling error in the causal messages on the error found in the probabilities computed for the convergence node is discussed.

4.1 The Posterior Convergence Error

The error that has arised in the probabilities computed for a convergence node of a Bayesian network in its prior state, may change in size as soon as an observation is entered into the network. An observation can influence the computed probabilities through causal messages or through diagnostic messages to the convergence node. An observation that affects the error through a causal message trivially changes the prior convergence error by conditioning all probabilities involved on the entered observation. An observation that affects the error through a diagnostic message, on the other hand, fundamentally changes the expression of

the convergence error. Below, we elaborate on the effect of a diagnostic observation. Consider a network composed of just a node C with a child F, and suppose that the observation $F = f$ is entered. Pearl's algorithm computes the posterior probabilities for node C from the compound diagnostic parameter $\lambda(C) = \Pr(f \mid C)$ and the compound causal parameter $\pi(C)$, using the rule

$$\Pr(C \mid f) = \alpha \cdot \lambda(C) \cdot \pi(C)$$

Normalisation then results in the posterior probability

$$\Pr(c \mid f) = \frac{\Pr(f \mid c) \cdot \pi(c)}{\Pr(f \mid c) \cdot \pi(c) + \Pr(f \mid \bar{c}) \cdot \big(1 - \pi(c)\big)}$$

for the value c of C.

Now consider the network from Figure 1 and suppose that the observation $F = f$ is entered into the network. Pearl's algorithm computes the compound causal parameter $\widetilde{\pi}(C)$ for the convergence node C using the causal messages $\widetilde{\pi}_C(A)$ and $\widetilde{\pi}_C(B)$. After an observation has been entered for node F, these messages may include a cycling error. In determining the posterior convergence error, the cycling error is left out of consideration and it is assumed that node C receives the correct causal messages $\pi_C(A)$ and $\pi_C(B)$. From these exact causal messages node C would compute a compound causal parameter that includes just a convergence error. This causal parameter will be indicated with $\widetilde{\pi}_{conv}(c)$[2]. The posterior probability of c given f that would be established from $\widetilde{\pi}_{conv}(c)$ will be indicated by $\widetilde{\Pr}_{conv}(c \mid f)$.

Recall that in the graphical illustration of the prior convergence error for node C in Figure 3, the depicted line segment captures, for the network from Figure 1, the exact probability $\Pr(c)$ as a function of $\Pr(d)$, given the conditional probabilities for nodes A, B and C; note that $\Pr(c) = \pi(c)$. The surface captures the approximate probability $\widetilde{\Pr}(c)$ as a function of $\pi_C(a)$ and $\pi_C(b)$, given the conditional probabilities for node C; note that $\widetilde{\Pr}(c) = \widetilde{\pi}_{conv}(c)$. When the transformation that is defined by $\Pr(c \mid f) = \frac{\Pr(f|c) \cdot \pi(c)}{\Pr(f|c) \cdot \pi(c) + \Pr(f|\bar{c}) \cdot \left(1 - \pi(c)\right)}$ is applied to the surface and the line segment, Figure 4 results. The line segment now captures $\Pr(c \mid f)$ and the surface now captures $\widetilde{\Pr}_{conv}(c \mid f)$. As for the prior state of the network, the approximate probability $\widetilde{\Pr}_{conv}(c \mid f)$ is found as the orthogonal projection of the exact probability $\Pr(c \mid f)$ on the surface. By subtracting $\widetilde{\Pr}_{conv}(c \mid f)$ from $\Pr(c \mid f)$, an expression is derived for the posterior convergence error found in the probability for c given f:

$$\Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f) = \frac{\Pr(f \mid c) \cdot \pi(c)}{\Pr(f \mid c) \cdot \pi(c) + \Pr(f \mid \bar{c}) \cdot \pi(\bar{c})} - \frac{\Pr(f \mid c) \cdot \widetilde{\pi}_{conv}(c)}{\Pr(f \mid c) \cdot \widetilde{\pi}_{conv}(c) + \Pr(f \mid \bar{c}) \cdot \widetilde{\pi}_{conv}(\bar{c})}$$

Rearranging terms gives

$$\Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f) = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c}) \cdot v}{\Pr(f) \cdot \Big(\Pr(f) - \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) \cdot v\Big)}$$

In the sequel u is used to denote the posterior convergence error $\Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f)$.

Below some of the characteristics of the function u are discussed. First of all, the sign of the posterior convergence error u equals the sign of the prior convergence error v because the sign of the numerator of the expression for u equals the sign of v and the sign of the denominator of the expression for u is always positive. The latter because $|\Pr(f \mid c) - \Pr(f \mid \bar{c})| \leq \Pr(f)$ and $v \in [-0.5, 0.5]$. From the numerator of the first derivative of u with respect to v

$$\frac{\partial u}{\partial v} = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c})}{\Big((\pi(c) - v) \cdot \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) + \Pr(f \mid \bar{c})\Big)^2}$$

being positive, moreover, it follows that, all other factors being constant, u increases with increasing v. Furthermore, given that $\Pr(f) \neq 0$, the posterior convergence error equals zero for $\Pr(f \mid c) = 0$, $\Pr(f \mid \bar{c}) = 0$ or $v = 0$. For $\Pr(f \mid c) = 0$, we have that $\widetilde{\Pr}_{conv}(c \mid f)$, is independent of the causal parameter

[2]Note that $\widetilde{\pi}_{conv}(c)$ equals the compound causal parameter $\tilde{\pi}(c)$ of the prior network but in general does not equal $\tilde{\pi}(c)$ of the posterior network.

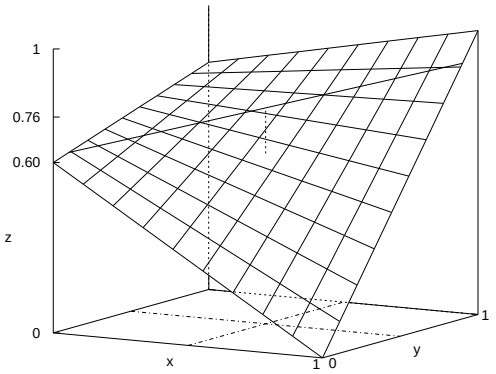


Figure 5: The line capturing $\Pr(c)$ and the surface capturing $\widetilde{\Pr}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ for the network from Figure 2.

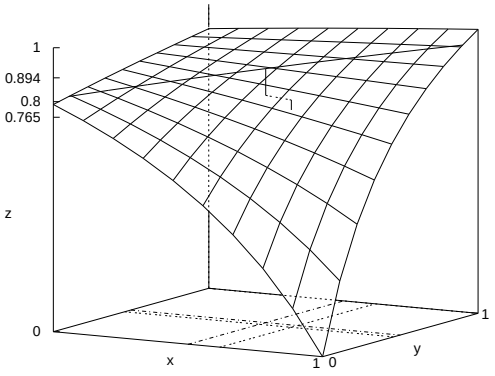


Figure 6: The line capturing $\Pr(c \mid f)$ and the surface capturing $\widetilde{\Pr}_{conv}(c \mid f)$ and $\widetilde{\Pr}(c \mid f)$ as a function of $x = \pi_C(a)$ c.q. $\widetilde{\pi}_C(a)$ and $y = \pi_C(b)$ c.q. $\widetilde{\pi}_C(b)$ for the network from Figure 2.

$\widetilde{\pi}_{conv}(c)$, and therefore, an error in this parameter will not affect $\widetilde{\Pr}_{conv}(c \mid f)$. A similar observation holds for $\Pr(f \mid \bar{c}) = 0$. For $v = 0$, no error is present in $\widetilde{\pi}_{conv}(c)$, and thus obviously, $\widetilde{\Pr}_{conv}(c \mid f)$ indeed equals $\Pr(c \mid f)$. Furthermore, for $\Pr(f \mid c) = \Pr(f \mid \bar{c})$, we have that u equals v. This is not surprising because in that case C and F are independent.

The expression of the posterior convergence error is ill-defined for the combination $\pi(c) = 0$ and $\Pr(f \mid \bar{c}) = 0$ and for the combination $\pi(c) = 1$ and $\Pr(f \mid c) = 0$. Note that those combinations both imply that $\Pr(f) = 0$. Below the limit of u for $\pi(c) = 0$ and $\Pr(f \mid \bar{c}) \downarrow 0$ is established, assuming that $v \neq 0$. For $\pi(c) = 0$, is found that

$$u = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c}) \cdot v}{\Pr(f \mid \bar{c})^2 - \Pr(f \mid \bar{c}) \cdot \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) \cdot v}$$

division by $\Pr(f \mid \bar{c})$ gives

$$u = \frac{\Pr(f \mid c) \cdot v}{\Pr(f \mid \bar{c}) - \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) \cdot v}$$

and taking $\Pr(f \mid \bar{c}) = 0$ gives

$$\lim_{\Pr(f \mid \bar{c}) \downarrow 0} = -1$$

The limit for $\pi(c) = 1$ and $\Pr(f \mid c) \downarrow 0$ and $v \neq 0$ can be established analogously to equal 1. For the range of the posterior convergence error thus is found that $u \in \langle -1, 1 \rangle$. Note that for the combinations $\pi(c) = 0$, $\Pr(f \mid \bar{c}) \downarrow 0$ and $\pi(c) = 1$, $\Pr(f \mid c) \downarrow 0$, extremely small probabilities $\Pr(f)$ are found. The posterior convergence error thus reaches its extreme values for extreme unlikely observations $F = f$.

4.2 The Effect of the Cycling Error

Consider a network with just a single simple loop. As argued in Section 2.2, the approximation of a probability for a convergence node given an observation for one of its descendents may not just include a convergence error. Given that for all convergence nodes in the loop, the convergence node itself or one of its descendants is observed, information may cycle in the loop and the causal messages of the parents of the convergence nodes may include a cycling error. As a result, a cycling error may be included in the compound causal parameters computed for the convergence nodes which in turn may introduce a cycling error in the approximations for the convergence nodes. The additional error is illustrated in Figures 5 and 6.

Figure 5 shows graphically the prior error $\Pr(c) - \widetilde{\Pr}(c) = \pi(c) - \widetilde{\pi}(c)$ found in the network from Figure 2. Again, the line segment captures the exact probabilities $\Pr(c)$ (which equal the exact compound parameters $\pi(c)$) as a function of $\Pr(d)$ given probabilities specified for A, B and C. For the $\Pr(d) =$

0.5 from the example network is found that $\pi(c) = 0.76$. The surface, again, captures the approximate probabilities $\widetilde{\Pr}(c)$ (which equal the approximate compound parameters $\widetilde{\pi}(c)$) as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ given the probabilities specified for node C. For the $\pi_C(a) = 0.5$ and $\pi_C(b) = 0.5$ from the example network, is found that $\widetilde{\pi}(c) = 0.6$. The prior convergence error equals $0.76 - 0.6 = 0.16$. Figure 6 shows, for the same network, the situation after the observation $F = f$ has been entered. The curve segment now captures $\Pr(c \mid f)$ as a function of $\Pr(d)$. For $\Pr(d) = 0.5$, is found that $\Pr(c \mid f) = 0.894$. The surface captures $\widetilde{\Pr}_{conv}(c \mid f)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ given the specified conditional probabilities of C. The exact causal messages $\pi_C(a) = 0.5$ and $\pi_C(b) = 0.5$ result in the compound causal parameter with just a convergence error $\widetilde{\pi}_{conv}(c) = 0.6$ and in the approximation with just a convergence error $\widetilde{\Pr}_{conv}(c \mid f) = 0.8$. The surface also captures $\widetilde{\Pr}(c \mid f)$ as a function of $\widetilde{\pi}_C(a)$ and $\widetilde{\pi}_C(b)$. The causal messages include a cycling error and equal $\widetilde{\pi}_C(a) \approx 0.6180$ and $\widetilde{\pi}_C(b) \approx 0.4607$ resulting in the actual compound causal parameter $\widetilde{\pi}(c) = 0.5491$. This actual parameter results in the actual approximation $\widetilde{\Pr}(c \mid f) \approx 0.765$. In terms of Figure 6, due to the cycling error in the causal messages, the actual approximation is not found as the orthogonal projection of the exact probability, but is shifted to another spot on the surface. In the network from Figure 2 the total error u equals approximately $0.894 - 0.765 = 0.129$.

Experiments showed that the point $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b))$ is located on the line that is defined by the point with the exact causal messages $(\pi_C(a), \pi_C(b))$ and the orthogonal projection of the saddle point of the surface on the base (SP). This latter point equals $(\frac{\Pr(c|\bar{a}\bar{b}) - \Pr(c|\bar{a}b)}{\Pr(c|ab) - \Pr(c|a\bar{b}) - \Pr(c|\bar{a}b) + \Pr(c|\bar{a}\bar{b})}, \frac{\Pr(c|\bar{a}\bar{b}) - \Pr(c|a\bar{b})}{\Pr(c|ab) - \Pr(c|a\bar{b}) - \Pr(c|\bar{a}b) + \Pr(c|\bar{a}\bar{b})})$. When the surface is a plane, that is, when $\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b}) = 0$, the slope of the surface determines the line on which the approximations will be found. The line then has the highest gradient.

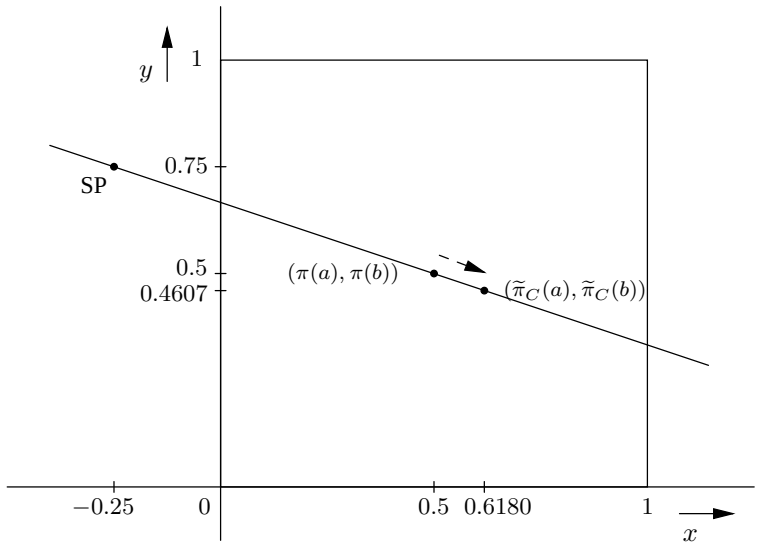


Figure 7: The line on which the approximate causal messages $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b))$ from nodes A and B to node C of the example network from Figure 2 are located given the observation $F = f$. The line is defined by the orthogonal projection on the base of the saddle point of the surface with the approximate probabilities of c given f, SP=$(-0.25, 0.75)$, and by the exact causal messages, $(\pi_C(a), \pi_C(b)) = (0.5, 0.5)$.

Example 4.1 *For the example network from Figure 2, the equation of the line on which the approximate causal messages are located equals*

$$y = -\frac{1}{3} \cdot x + \frac{2}{3}$$

The line is depicted in Figure 7. The point with the approximate causal messages found upon loopy propagation $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b)) \approx (0.6180, 0.4607)$ *indeed is found on the line defined by the projection of the saddle point of the surface SP*$= (-0.25.0.75)$ *and the exact causal messages* $(\pi_C(a), \pi_C(b)) = (0.5, 0.5)$.

5 Conclusions

During loopy propagation, errors arise in the loop nodes of a network. Cycling errors arise at the inner nodes a co-occur with the cycling of information and convergence errors arise at the convergence nodes due to a disregard of dependencies. In a network in its prior state, only convergence errors may be introduced. In previous research, we identified the factors that govern the size of the convergence error in the probabilities computed for the convergence nodes. In this paper, we studied the error found in the probabilities computed for the convergence node of a binary network with just a simple loop in a posterior state. After a causal observation, as in the prior state, only a convergence error may be found and this error is governed by essentially the same factors as the convergence error in the network in its prior state. A diagnostic observation, however, fundamentally changes the expression of the convergence error. The posterior convergence error may be large, approaching 1, where the maximum size of the prior convergence error is 0.5. The extremes for the posterior convergence error are reached given unlikely observations. After a diagnostic observation, moreover, a cycling error may enter the probabilities computed for the convergence node by the causal messages from its the parents. The additional effect of the cycling error on the computed probabilities was illustrated graphically.

Acknowledgments
The research reported in this thesis 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, in cooperation with the IEEE computer society, Luxembourg.

[2] J.H. Bolt (2008). *Bayesian Networks: Aspects of Approximate Inference.* PhD thesis, Department of Information and Computing Sciences. Utrecht University.

[3] J.H. Bolt (2008). Bayesian networks: the parental synergy. To appear in: *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models.*

[4] G.F. Cooper (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, **42**, 393 – 405.

[5] P. Dagum, M. Luby (1993). Approximate inference in Bayesian networks is NP hard. *Artificial Intelligence*, **60**, 141 – 153.

[6] K. Murphy, Y. Weiss, M. Jordan (1999). Loopy belief propagation for approximate inference: an empirical study. In: K.B. Laskey, H. Prade (editors), *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, 467 – 475.

[7] J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Palo Alto.

[8] Y. Weiss (2000). Correctness of local probability propagation in graphical models with loops. *Neural Computation*, **12**, 1 – 41.

[9] Y. Weiss, W.T. Freeman (2001). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, **13**, 2173 – 2200.

Automatic Thesaurus Generation using Co-occurrence

Rogier Brussee [a] Christian Wartena [a]

[a] *Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands*

Abstract

This paper proposes a characterization of useful thesaurus terms by the informativity of co-occurrence with that term. Given a corpus of documents, informativity is formalized as the information gain of the weighted average term distribution of all documents containing that term. While the resulting algorithm for thesaurus generation is unsupervised, we find that high informativity terms correspond to large and coherent subsets of documents. We evaluate our method on a set of Dutch Wikipedia articles by comparing high informativity terms with keywords for the Wikipedia category of the articles.

1 Introduction

We consider the problem of generating a thesaurus for a given collection of texts using statistical methods. More precisely we try to produce a list of terms which are the most informative for understanding the collection as a whole (i.e. only a controlled vocabulary). This problem is related to, but different from, assigning keywords to a text from a list of keywords, and that of finding the most characteristic terms for a given subset of the corpus. Part of the attraction of the current approach is that it proposes a statistical model to formalize the notion of a (good) keyword. If we believe in the assumptions leading to this model, the high level algorithms to generate a thesaurus are almost forced upon us.

Our main assumption is that co-occurrence of terms is a proxy for their meaning [12, 15, 10]. To use this information, we compute for each term z the distribution of all co-occurring terms $\bar{p}_z(t)$. We can then use this co-occuring term distribution as a proxy for the meaning of the term in the context of the collection and compare it with the term distribution of a single document. We assume that a document d is semantically related to a term z if the term distribution $q_d(t)$ of the document is similar to the co-occurring term distribution $\bar{p}_z(t)$ of z. Fortunately, there is a natural similarity measure for probability distributions, the relative entropy or Kullback-Leibler divergence. If we follow this formalization through, there is a straightforward strategy for generating a thesaurus as the set of terms which give the greatest overall information gain, defined as a difference of Kullback-Leibler divergences. In practice this model turns out to be an oversimplification, mainly because the same subject can be characterized by different terms, which produces somewhat disappointing results. We will discuss this in section 5.

The organization of this paper is as follows, in section 2 we discuss some related work. In section 3 we introduce the different probability distributions and the information theoretic notion of Kullback-Leibler divergence that will be the basis for the rest of the paper. We use these in section 4 to give various definitions of information gain that can be used to rank keywords. In section 5 we evaluate our notions on a corpus of the Dutch Wikipedia.

2 Related work

The problem of finding the set of thesaurus terms to describe the documents of a collection is closely related to the problem of determining term weights for indexing documents. In particular it is related to approaches in which the weight of a term for a document is computed from its frequency in the document and its importance in the collection as a whole. One of the most elegant approaches is the term discrimination model of Salton (see [16, 20], and references cited there). Given a vector space model in which each

dimension represents one term, each document can be represented as a vector, the discrimination value of a term is the change of the average distance between all documents if the corresponding dimension is deleted from the model. Of course, various distance measures can be used to compute the distance between documents in this model. Conceptually, this approach is somewhat similar to ours: deleting a term with a high discrimination value causes a higher density, or lower entropy. However, we do not look at the effect of deleting a term, but on the compression that can be obtained by knowing that a term is contained in a document. Thus co-occurrence of terms plays a crucial role in our approach, while this is not taken into account in the computation of the discrimination value.

Another approach to the problem of finding a set of thesaurus terms is pursued by [3]. They start clustering documents (based on distances in a term vector space model) and then try to find discriminative terms for the clusters.

The methods used in this paper are somewhat similar to latent semantic analysis [5, 10] since both have co-occurrence matrices of terms and documents as their starting point. While the current approach also naturally leads to combinations of keywords, it has the major advantage of leading only to a probability distribution over keywords (i.e. weighted sums of keywords with total weight 1) rather than positive and negative combinations. It is also based on conceptually similar, but technically different notion of proximity between keywords that has a more direct information theoretic interpretation. Unfortunately, like the singular value decomposition needed for latent semantic analysis, it is computationally heavy. Our method is even closer to, and influenced by the probabilistic latent semantic analysis (PLSA) of [6, 7]. Like PLSA our method is based on maximizing Kullback-Leibler divergence and maximal likelihood. However unlike Hofmann's work we try to find probability distributions over terms rather than some assumed underlying abstract aspects of the text that "explain" the observed distribution of terms in documents.

A lot of work has also been done on the assignment of keywords to individual documents or to subsets of a corpus of documents. As examples for this field of research we refer to [1, 8] and to [18] who focus on finding keywords in different sections of biomedical articles.

There is a vast literature on the information theoretic background of this paper. We used [2] as our main information theoretic reference, which we found a quite readable introduction.

3 Term and document distributions

We simplify a document to a bag of words or terms[1]. Once we accept this model, the number of occurrences of all the different terms in all the different documents is the only information we have left. Thus, consider a set of n term occurrences $\mathcal{W}$ each being an instance of a term t in $\mathcal{T} = \{t_1, \ldots t_m\}$, and each occurring in a source document d in a collection $\mathcal{C} = \{d_1, \ldots d_M\}$. Let $n(d,t)$ be the number of occurrences of term t in d, $n(t) = \sum_d n(d,t)$ be the number of occurrences of term t, and $N(d) = \sum_t n(d,t)$ the number of term occurrences in d.

3.1 Basic Distributions

We define probability distributions $\mathbf{Q}$ on $\mathcal{C}\times\mathcal{T}$, a distribution Q on $\mathcal{C}$ and q on $\mathcal{T}$ that measure the probability to randomly select a term occurrence, and the corresponding term, or source document.

$$\begin{aligned} \mathbf{Q}(d,t) &= n(d,t)/n && \text{on } \mathcal{C}\times\mathcal{T} \\ Q(d) &= N(d)/n && \text{on } \mathcal{C} \\ q(t) &= n(t)/n && \text{on } \mathcal{T} \end{aligned}$$

These distributions are the baseline probability distributions for everything that we will do in the remainder and we will use them in favor of the simple counting measure when determining the "size" of subsets of $\mathcal{C}$ and $\mathcal{W}$. In addition we have two important conditional probability distributions

$$\begin{aligned} Q(d|t) = Q_t(d) &= n(d,t)/n(t) && \text{on } \mathcal{C} \\ q(t|d) = q_d(t) &= n(d,t)/N(d) && \text{on } \mathcal{T} \end{aligned}$$

We use the notation $Q(d|t)$ for the *source distribution of* t, as it is the probability that a random term occurrence has source d given that we know that that it is an instance of term t. Similarly, $q(t|d)$, the *term*

[1] In the experiments discussed in section 5 we will do some preprocessing, like lemmatization and multiword detection. Thus we use terms rather than words

distribution of d is the probability that a random term occurrence is an instance of term t given that the source is d. Other probability distributions on $\mathcal{C} \times \mathcal{T}$, $\mathcal{C}$ and $\mathcal{T}$ will be denoted by $\mathbf{P}$, P, p with various sub and superscripts .

3.2 Distribution of Co-occurring Terms

Consider a Markov chain on $\mathcal{T} \cup \mathcal{C}$ having transitions $\mathcal{C} \to \mathcal{T}$ with transition probabilities $Q(d|t)$ and transitions $\mathcal{T} \to \mathcal{C}$ with transition probabilities $q(t|d)$ only. The chain allows us to propagate probability distributions from terms to document and vice versa.

Given a term distribution $p(t)$ we compute the one step Markov chain evolution. This gives us a document distribution $P_p(d)$, the probability to find a term occurrence in a particular document given that the term distribution of the occurrences is p

$$P_p(d) = \sum_t Q(d|t)p(t).$$

Likewise given a document distribution $P(d)$, the one step Markov chain evolution is the term distribution $p_P(t) = \sum_d q(t|d)P(d)$. Since $P(d)$ gives the probability to find a term occurrence in document d, p_P is the P-weighted average of the term distributions in the documents. Combining these, i.e. running the Markov chain twice, every term distribution gives rise to a new distribution

$$\bar{p}(t) = p_{P_p}(t) = \sum_d q(t|d)P_p(d) = \sum_{t',d} q(t|d)Q(d|t')p(t')$$

In particular starting from the degenerate "known to be z" term distribution $p_z(t) = p(t|z) = \delta_{tz}$ (1 if $t = z$ and 0 otherwise), we get the *distribution of co-occurring terms* $\bar{p}_z$

$$\bar{p}_z(t) = \sum_{d,t'} q(t|d)Q(d|t')p_z(t') = \sum_d q(t|d)Q(d|z).$$

This distribution is the weighted average of the term distributions of documents containing z where the weight is the probability $Q(d|z)$ that an instance of term z has source d.

Note that the probability measure $\bar{p}_z$ is very similar to the setup in [11, section 3]. However, we keep track of the density of a keyword in a document rather than just the mere occurrence or non occurrence of a keyword in a document. This difference is particularly relevant for a long document in which a term occurs with low density, because it has a relatively high contribution to the mean word distribution. Unfortunately $\bar{p}_z$ is expensive to compute.

4 Informativity of Keywords

Intuitively, a keyword makes it easier to remember the content of a document given some knowledge of similar documents. Formalizing this intuition, we base a criterion for the informativity of a term on the Kullback-Leibler divergence (KL-divergence) between the mean distribution of terms co-ocurring with a given keyword with the term distribution of the whole collection.

We quickly recall the definition, basic properties and interpretation of the KL-divergence [2, sec. 2.3]. Given a finite set of symbols $X = \{x_1, x_2, \ldots, x_n\}$ with two probability distributions $p(x_i) = p_i$ and $q(x_i) = q_i$ the KL-divergence is defined as

$$D(p||q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right)$$

It is easy to show that $D(p||q) \geq 0$ with equality iff $p_i = q_i$ for all i.

The KL-divergence has the following standard interpretation. In an optimal compression scheme, a stream of symbols $(x_{i_1}, x_{i_2}, \ldots)$ (with $1 \leq i_k \leq n$) over the alphabet X distributed according to a probability distribution p uses at least $\log(p_i^{-1})$ bits for the symbol x_i. Thus with optimal compression we need on *average* $\sum_{i=1}^{n} p_i \log(p_i^{-1})$ bits per symbol. If the stream is compressed using a scheme adapted to a distribution q instead, we use on average $\sum_{i=1}^{n} p_i \log(q_i^{-1})$ bits per symbol. Therefore, the KL-divergence is the average number of bits saved per symbol by using the actual probability distribution p of the stream rather than some *a priori* distribution q.

4.1 Information Gain for a document

Given a collection of documents and a term, there is a subcollection for which the term is relevant. To determine this subcollection we will define an information theoretic measure defined as average number of bits saved by using a specialized compression scheme for the subcollection compared to using a single scheme for the whole collection.

The net information gain of using a term distribution p for a document d is defined as

$$\begin{aligned} IG(d,p) &= D(q_d||q) - D(q_d||p) \\ &= \sum_{t,q_d(t)\neq 0} q_d(t) \log\left(\frac{p(t)}{q(t)}\right) \end{aligned}$$

It measures the gain or loss of compression obtained from using p compared to the corpus term distribution q. We will call this the *specific net information gain of using* p to emphasize that this is a gain per word. Clearly $IG(d,p) \leq D(q_d||q)$, and the unique maximum is attained for $p = q_d$. It is possible that $IG(d,p) = -\infty$, but $IG(d,p)$ is finite if and only if $p(t) = 0$ implies $q_d(t) = 0$, since $q(t) = 0$ certainly implies $q_d(t) = 0$.

4.2 Information Gain for a Subcollection

For a subcollection of documents $\mathcal{D} \subset \mathcal{C}$, we define the *fractional net specific information gain* by weighing the specific information gain with the size of the document relative to the collection

$$\overline{IG}(\mathcal{D},p) = \sum_{d\in\mathcal{D}} Q(d)\, IG(d,p) = \sum_{d\in\mathcal{D},t\in\mathcal{T}} Q(d) q_d(t) \log\left(p(t)/q(t)\right)$$

Defining the fraction of term occurrences in $\mathcal{D}$, $\beta_{\mathcal{D}} = \sum_{d\in\mathcal{D}} Q(d)$ and the average term distribution $p_{\mathcal{D}} = \beta_{\mathcal{D}}^{-1} \sum_{d\in\mathcal{D}} Q(d) q_d$, of documents in the subcollection $\mathcal{D}$, we rewrite $\overline{IG}(\mathcal{D},p)$ to

$$\overline{IG}(\mathcal{D},p) = \beta_D \sum_{t\in\mathcal{T}} p_{\mathcal{D}}(t) \log\left(p(t)/q(t)\right) = \beta_{\mathcal{D}}\left(D(p_{\mathcal{D}}||q) - D(p_{\mathcal{D}}||p)\right)$$

We define the *fractional specific gain of the subcollection* by setting $p = p_{\mathcal{D}}$. This clearly maximizes $\overline{IG}(\mathcal{D},p)$ and we have

$$\overline{IG}(\mathcal{D}) = \overline{IG}(\mathcal{D},p_{\mathcal{D}}) = \beta_{\mathcal{D}} D(p_{\mathcal{D}}||q) \geq 0. \tag{1}$$

4.3 Information Gain for a Keyword

Consider the subset $\mathcal{C}_z$ of documents benefitting from compression against the average distribution $\bar{p}_z$ of terms co-occurring with z i.e. $\mathcal{C}_z = \{d \in \mathcal{C} |\, IG(d|\bar{p}_z) > 0\}$. We then have a natural measure for the informativity of using z as a key term. We define $\overline{IG}_z^+$, the *fractional net specific information gain of a key term* z as the non negative number

$$\overline{IG}_z^+ = \overline{IG}(\mathcal{C}_z|\bar{p}_z) = \sum_{d\in C_z} Q(d)\, IG(d|\bar{p}_z).$$

Note that a document d might be in C_z even if there is no occurrence of z in d! It is also possible that $\overline{IG}(\mathcal{C}_z) > \overline{IG}_z^+$, because it it need not be the case that $\bar{p}_z = p_{C_z}$.

5 Evaluation

We have implemented a number of techniques to extract a thesaurus from a collection of texts. All tested algorithms give a ranking of all detected terms.

To compare different strategies we compiled a small corpus of Dutch Wikipedia articles consisting of 758 documents. In the analysis phase, 118099 term occurrences, and 26373 unique terms were found. The articles were taken from 8 Wikipedia categories: spaceflight, painting, architecture, trees, monocots, aviation, pop music, charadriiformes. Categories were selected for subjective similarity, like spaceflight and aviation, and subjective dissimilarity like pop music and monocots. Articles are equally distributed over

the categories, but the average length, homogeneity and specificity of articles differs significantly between categories. This is clearly reflected in the information gain obtained by splitting the collection in the articles from one category and from the remaining categories and using formula (1) in section 4.2. The results are given in Table 1. One reason for the high score of the category *pop music* is the fact that the Dutch articles on this subject contain many lists of English song titles, whereas in the other categories English words are rather rare.

spaceflight	0.31	monocots	0.19	painting	0.40	aviation	0.38
architecture	0.24	pop music	0.47	trees	0.31	charadriiformes	0.20

Table 1: Information gain for all document categories

5.1 Preprocessing

Before extracting thesaurus terms we do some preprocessing using the GATE–framework [4]. The main analysis steps are: lemmatization, multiword lookup and named entity recognition. Lemmatization and tagging is done using the Treetagger [17]. Tagging allows us to distinguish content words from function words. We only use content words. After lemmatization all inflected forms of verbs, nouns and adjectives are mapped onto their lexical form, substantially reducing the variation in the input data. Note that we used lemmatization, not stemming. Unlike stemming lemmatization distinguishes between different words with the same root, like *production* and *product*. In case of ambiguity the correct form is chosen by a trained statistical model. Since adjectives are seldom used as a keyword we have filtered out the adjectives from the result lists in all experiments, even though adjectives are bearers of relevant semantic information, e.g. in a phrase like *chemical process*. For multiword lookup we used article titles from the Dutch Wikipedia, since it is reasonable to assume that each title represents a single concept. Finally, some simple rules are applied to identify proper names. While some of the components are language dependent, all of the components are available for a number of languages within the GATE–framework.

As an additional filter we require that a potential thesaurus term z has to be specific enough. It turned out that there are some highly frequent words that have a high $\overline{IG}_z^+$ but that cannot be correlated to any subject. We therefore require that $\bar{p}_z$ be different enough from the background distribution q, as measured by the Kullback-Leibler divergence. We used a cutoff $D(\bar{p}_t||q) > 1$ bit, that turned out to give decent results.

5.2 Thesaurus construction

The resulting list of terms is now ranked based on the value of $\overline{IG}^+$. In fact we got slightly better results by slightly modifying $\overline{IG}^+$ and using a naively smoothed version of the co-occurrence distribution $\bar{p}_t^{(sm)} = 0.95\bar{p}_t + .05q$. Smoothing also simplified software development, as the KL-divergence of a distribution with respect smoothed distribution is automatically finite.

Since we were not aware of other research on thesaurus extraction without prior knowledge of categories, we compared the $\overline{IG}^+$ ranking with the ranking given by the discrimination values for term weighting from [16] and some more simple and naive rankings.

For the computation of discrimination values (DV) we have implemented the algorithm from [20] using cosine dissimilarity. For our corpus we found that this method strongly favors terms specific for one or a few documents, especially large ones. This method might be suited to find discriminating terms for a specific document, but seems to be less suited for identifying useful global keywords.

The first "naive" method we evaluated is to take the most frequent words. This approach gives reasonable results probably because we only consider nouns, verbs (without auxiliaries) and proper names. The last method is to compute the average *tf.idf* value over all documents for each term. We use the following formula to compute *tf.idf* for a corpus C, a term t and a document $d \in C$: $tf.idf(d,t) = (1 + \log n(d,t)) \cdot M/df(t)$ where M is the number of documents in C and $df(t) = \Sigma_d m(t,d)$ where $m(t,d) = 1$ if $n(t,d) > 0$ and $m(t,d) = 0$ otherwise (see e.g. [14]).

5.3 Results

To evaluate the resulting generated thesaurus we determine (a) the number of the thesaurus terms that are really informative and (b) the number of topics in the data set that are covered.

category	*keywords*	*category*	*keywords*
spaceflight	aarde, lancering, lanceren, raket, satelliet	painting	werk, werken[2], Van Gogh, schilderen, schilderij
architecture	gebouw, huis, station, bouwen, kerk	trees	boom, blad, hout, vrucht, gebruiken
monocots	familie, plant, soort, geslacht, eenzaadlobbig	aviation	vliegtuig, toestel, vliegen, motor, Schiphol
pop music	album, band, the, single, nummer	charadriiformes	vogel, soort, geslacht, zwart, kust

Table 2: Keywords for the document categories.

To evaluate (a) (precision) we have compared the generated thesaurus terms with a broad thesaurus of keywords used by the Dutch Institute of Sound and Vision, the Gemeenschappelijke Thesaurus voor Audiovisuele Archieven (GTAA, Common Thesaurus for Audiovisual Archives), containing about 9000 subject terms and extensive lists of person names, company names and geographical names [13]. The fraction of the keywords found in this thesaurus was used as an indication of the quality of the set of automatically generated thesaurus terms.

Determining (b) (recall) presupposes a knowledge of the topics in the corpus. Of course the full list of "topics" is open to debate, but it seems reasonable to count each of the eight selected Wikipedia categories as a topic. To get an impression of the coverage of these topics we have extracted the five best keywords for each category resulting in a set of 30 terms. A keyword selection method is considered better if more of these terms are found. To find the best keywords for each category we have used a slight variations of the method proposed by [1]: we compare the term distribution of the whole corpus with the term distribution of the subset and select those terms that have the most deviating distributions. The resulting sets of keywords are given in Table 2.

The results for the coverage of the best n terms by the GTAA is shown in Table 3 (left). We have determined the coverage for n is 40, 80 and 160. The low percentages are partially due to the fact that

n	*DV*	*freq*	*tf.idf*	$\overline{IG}^+$
40	0,18	0,33	0,38	0,48
80	0,16	0,39	0,35	0,40
160	0,18	0,43	0,34	0,38

n	*DV*	*freq*	*tf.idf*	$\overline{IG}^+$
40	1 (1)	16 (8)	11 (4)	12 (7)
80	1 (1)	28 (8)	13 (5)	15 (8)
160	1 (1)	33 (8)	20 (8)	20 (8)

Table 3: Left table: Coverage by GTAA. Right table: recall of keywords (categories)

many of the generated thesaurus terms would not be considered good keywords (neither intuitively nor in the GTAA) although they do clearly relate to a specific domain e.g. 'guitar player' or 'French'. For the $\overline{IG}^+$-method we see a slight tendency to give higher ranks to the more meaningful terms. The results of our method are clearly better than the average *tf.idf* and at the low levels slightly better than the term frequency method.

To evaluate the recall of the eight subjects from which the texts were taken we can either count the number of keywords from Table 2 or the number of categories from which a keyword (according to the same table) is present in the suggested list. The results for the sets of the 40, 80 and 160 best terms according to each method are given in Table 3 (right). Surprisingly, the simple term frequency gives the best results for this evaluation measure. At the low levels the method presented in this paper is clearly better than the average *tf.idf*. On closer inspection of the generated terms one reason for the somewhat disappointing result becomes immediately clear: Our method tends to select words that have many co-occurring words with a similar distribution. This implies that all these co-occurring words are found as well. Thus, most of the best ranked terms are near synonyms, or at least terms on the same subject, suppressing the less prominent subjects. Clustering of terms with similar distributions (e.g. using a symmetrized KL-divergence as a similarity measure) would be a solution to this problem. However, results depend strongly on the clustering method and are difficult to compare to the other methods.

[2]Note that we used *lemmatization* differentiating between the noun *werk* (labor, work) and the verb *werken* (to work).

6 Conclusion

We developed an information theoretic measure $\overline{IG}^+$ to rank terms for their usefulness as keyword, based on the co-occurrence of a term with other terms. The measure gives a natural balance between specificity and generality of a term, and detects large and coherent subsets of a document collection that can be characterized by a term. To generate a thesaurus for a given corpus we used the highest ranked terms after some filtering. We proposed to select the highest ranked terms after some filtering as a basic word list to construct a thesaurus. Evaluation showed that the proposed method was better that the widely used *tf.idf* measure, but is not better than simpy selecting the most frequent words, again after some filtering. However, our method also allows a natural definition of a similarity measure between terms that can be used to cluster terms. If hierarchical clustering is used relations between (clusters of) words can be established as a further step in thesaurus construction ([9]). In [19] we explored clustering using divergences of co-occurrence distributions as a distance measure.

Acknowledgements

We thank Wolf Huijsen for extracting the data from Wikipedia. The research for this paper was part of a MultimediaN project sponsored by the Dutch government under contract BSIK 03031.

References

[1] Miguel A. Andrade and Alfonso Valencia. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7):600–607, 1998.

[2] T. Cover and J. Thomas. *Elements of information theory*. John Wiley and Sons, Inc., 1991.

[3] Carolyn J. Crouch and Bokyung Yang. Experiments in automatic statistical thesaurus construction. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 77–88, New York, NY, USA, 1992. ACM.

[4] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[5] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285, New York, NY, USA, 1988. ACM.

[6] T. Hofmann. Probabilistic latent semantic analysis. In *UAI99: Uncertainty in artificial intelligence*, 1999.

[7] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, January 2001.

[8] Anette Hulth and Beáta Megyesi. A study on automatically extracted keywords in text categorization. In *ACL*. The Association for Computer Linguistics, 2006.

[9] Dino Ienco and Rosa Meo. Towards the automatic construction of conceptual taxonomies. In Il-Yeol Song, Johann Eder, and Tho Manh Nguyen, editors, *DaWaK*, volume 5182 of *Lecture Notes in Computer Science*, pages 327–336. Springer, 2008.

[10] T.K. Landauer, P.W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[11] Hang Li and Kenji Yamanishi. Topic analysis using a finite mixture model. *Inf. Process. Manage.*, 39(4):521–541, 2003.

[12] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behaviour Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[13] Véronique Malaisé, Luit Gazendam, and Hennie Brugman. Disambiguating automatic semantic annotation based on a thesaurus structure. In *Actes de la 14e conférence sur le Traitement Automatique des Langues Naturelles*, pages 197–206, 2007.

[14] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[15] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th conference on Computational linguistics*, pages 304–309, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[16] G. Salton, C.S. Yang, and C.T. Yu. A theory of term importannce in automatic text analysis. Technical Report TR 74-208, Department of Computer Science, Cornell University, July 1974.

[17] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994. unknown.

[18] Parantu K. Shah, Carolina Perez-Iratxeta, Peer Bork, and Miguel A. Andrade. Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, 4(20):1–9, 2003.

[19] Christian Wartena and Rogier Brussee. Topic detection by clustering keywords. In *DEXA Workshops*, pages 54–58. IEEE Computer Society, 2008.

[20] Peter Willett. An algorithm for the calculation of exact term discrimination values. *Inf. Process. Manage.*, 21(3):225–232, 1985.

A Modern Turing Test: Bot Detection in MMORPGs

Adam Cornelissen [a] Franc Grootjen [b]

[a] *ICIS, Radboud University Nijmegen, the Netherlands*
[b] *NICI, Radboud University Nijmegen, the Netherlands*

Abstract

Modern online multiplayer games have become increasingly popular with gamers all around the world. This applies in particular to the kind of games that can be played with hundreds to thousands of players simultaneously, the so called 'massively multiplayer online games', often simply referred to as MMORPGs.

In these games players play as a virtual character taking on the role of a knight, priest, mage or some other heroic character to defeat enemies, to complete tasks (widely known as 'quests') or to compete in battles with other players.

While doing this, players receive items (such as gold or potions), new equipment (such as swords, shields and armor) or increased experience (how well your character is able to do a certain task) as a reward for their effort.

Not everyone though plays according to the rules of the game. A multitude of ways to cheat in games exist. In this paper we will try to find a method to automatically detect a kind of cheating where players use programs to automate their actions; the use of game bots. The presented method will be validated using a small scale experiment of twenty-five players and the same amount of bots.

1 Introduction

Massively multiplayer online role-playing games (MMORPGs) is a genre of computer role-playing games (CRPGs) in which a large number of players interact with each other in a virtual world. MMORPGs are becoming increasingly popular with gamers around the world. Millions of players worldwide (see Figure 1) pay a monthly fee to reside in an online game environment to play together.

This billion dollar industry[1] is facing the threat of cheating players, in particular those using automated tools (bots).

The use of game bots wrecks the balance of power and economy in such games; game developers adjust their game to what they think is 'reasonable gameplay' for normal players. Bots, however, play differently: they can, for example, play extremely long or handle repetitive gameplay elements over and over again. It is also known that in-game items gathered by bots are sold on auction websites such as E-Bay, thus having an impact on not only the virtual, but also the real economy.

Most game companies tried to protect their games against botting by using *traffic encryption*. Unfortunately since the encryption code is executed client side, and therefore publicly available, these protection mechanisms only last for a couple of weeks. Members of hacker communities consider it a challenge to be the first one to distribute the new modified working bots.

1.1 Bots

Normally, a player character is operated by a human being who is playing the game. However, tools exist to let your character play automatically, without human interaction. A character not being operated by a real person but by a computer program is called a 'game bot' or simply a 'bot', which is an abbreviation for 'robot'.

[1] Revenues from U.S. online gaming services will increase from $1.1 billion in 2005 to more than $3.5 billion in 2009 [3]

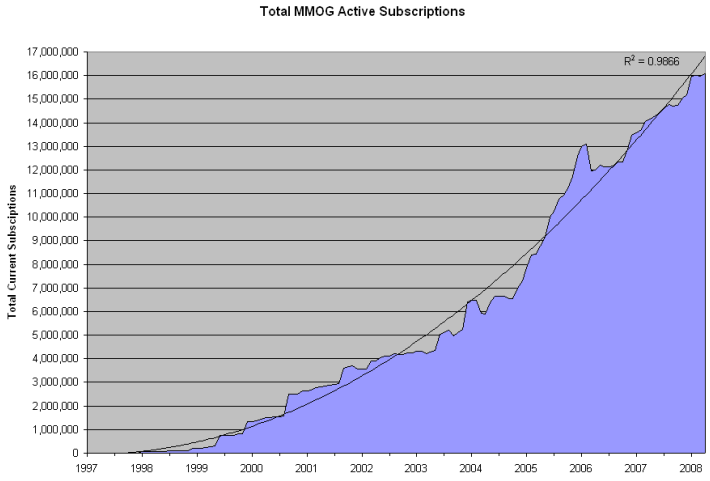


Figure 1: Total MMOG Active Subscriptions [9], these figures include *known* subscriptions to MMORPGs and do not include subscriptions to for example Ragnarok Online

1.2 Purpose of bots

Players advance in the game through completing quests, competing with other players and by defeating enemies. In the beginning every player has the weakest equipment in the game and very little experience at different actions (bad offense, bad defense, no skills such as item crafting, ...) but throughout the game players can gain better equipment, higher experience and more advanced skills.

To get higher experience players might choose to kill enemies. Certain types of enemies are often slain by using a similar strategy over and over again every time a player encounters such an enemy. While doing this, players gain experience which allows them to access newer and better game features.

People that enjoy playing the game but who simply lack the time to engage in the aforementioned repetitive gameplay might consider using bots for this repetitive and non-entertaining task.

Another reason why people use bots is to get better items and equipment. After slaying an enemy there is a chance that it will drop a certain item. Thus, if an item has a 0.01% chance to be dropped by a certain enemy, the item will drop approximately once per every 10.000 enemies slain. By using a bot, getting such an item is merely a matter of letting the bot run a long time instead of actually playing for hours.

Lack of time and/or lust to play the repetitive elements in the game and the will to easily get better ingame items and experience are the primary reasons for the use of bots.

1.3 Negative consequences of bots

At first glance it might seem that there is no reason for disallowing the usage of bots to handle the repetitive gameplay or to get items. However, there are some downfalls in allowing bots.

The use of bots can disrupts the balance of power in the game. When killing monsters over and over again (much more than one would normally do) the chances of getting a powerful item which rarely drops increases significantly.

Also, because bots can handle things repetitively and often very quickly, they can outplay human-controlled characters, giving them an unfair advantage over legitimate players.

For almost the same reason as disrupting the balance of power in the game, the use of bots can also break the economy of the game. In almost any game enemies can drop some kind of currency (cash, gold, ...). Letting a bot kill a lot of enemies gives the bot owner a lot of this currency, making him (a lot) wealthier than the average player. The process of letting a bot kill enemies to gain more ingame currency is very common and known by gamers as 'gold farming' or 'farming'.

Another reason to disallow bots is that they have a negative influence on the gaming experience of normal players. The use of bots is viewed as 'unfair' behaviour by the majority of the gaming community. Furthermore, since the game is multi user, players will encounter bots ingame. Obviously most bots are

antisocial, using shared game resources. These practices are known as 'kill stealing' and 'loot stealing'.

The most shocking fact about botting might be this: it is known that sometimes ingame items and currency gathered by bots are sold for real money on auction websites. This shows that bots do not only have an influence on the virtual economy, but also on the real economy.

For these (and other) reasons, companies running online games often disallow the use of bots in their Terms of Service (ToS) or End-User License Agreement (EULA).

1.4 Related work

Cheat prevention is regarded as a crucial challenge in the design of online games [1, 5, 6, 11]. Because game cheats often exploit loopholes in game rules or implementation bugs the main research focus is directed to correctness proofs and runtime verification of transaction atomicity [6]. Unfortunately, since bots do obey the rules of the game, these approaches do not apply to bot detection.

In a more recent study [4] Chen et al. describe an approach to identify MMORPG bots by analysing their traffic patterns. They propose strategies to distinguish bots from human players based on their traffic characteristics. The strategies are Command Timing, Traffic Burstiness, Reaction to Network Conditions. Unlike the approach presented in this paper, they do not use the *contents* of the traffic.

Yampolskiy and Govindaraju [10] as well as Golle and Ducheneaut [7] show how an embedded non-interactive test can be used to prevent bots from participating in online games. Although they report success in poker/card games, their embedded challenge response system appears to be difficult to integrate in MMORPGs without becoming annoying.

2 The Experiment

In order to create a detection method for bots, we decided to focus on 'Ragnarok Online' [8], a game known to be attacked by bots.

2.1 Ragnarok Online

Ragnarok Online, one of the most popular MMORPGs worldwide, was created by the Gravity Corporation in 2002. According to MMOGCHART [9]:

> *Ragnarok Online is supposedly the second biggest MMOG in South Korea, with well over 2 million subscribers. ... Recently they claimed 17 million worldwide with over 700,000 in North America, ...*

One reason for its popularity in Asia is the well-rendered anime style. Figure 2 shows a couple of players ingame. The game's design encourages players to get involved with other characters and form parties and guilds, which may be one of the main contributions to its popularity.

2.2 OpenKore

Unfortunately, due to its popularity there are several bot implementations for Ragnarok Online. The biggest well-known bot implementation is OpenKore, a free cross platform open source project [2]. According to the information on their own website the market share of OpenKore is about 95%. The software is highly configurable and (important for most users) runs out of the box. The availability and the easiness of use of OpenKore form the biggest threat to the game. OpenKore is the bot implementation that is used in our research. We will use the software as it is provided. This means that no changes will be made that will change the behaviour of the bot. It should be noted that the vast majority of people who use bots use it like this. The bot will automatically walk around, use teleporters, attack monsters, pick up items, sit down to recharge health, teleport back to town when very close to death and perform several other actions.

2.3 Collecting Data

Ragnarok Online consists of a client (running on the gamer's pc) which communicates with a server. The communication consists of a stream of packets which tells the server what actions the player performs. Using a network tool we were able to capture these packets, both for normal (human) players using the official

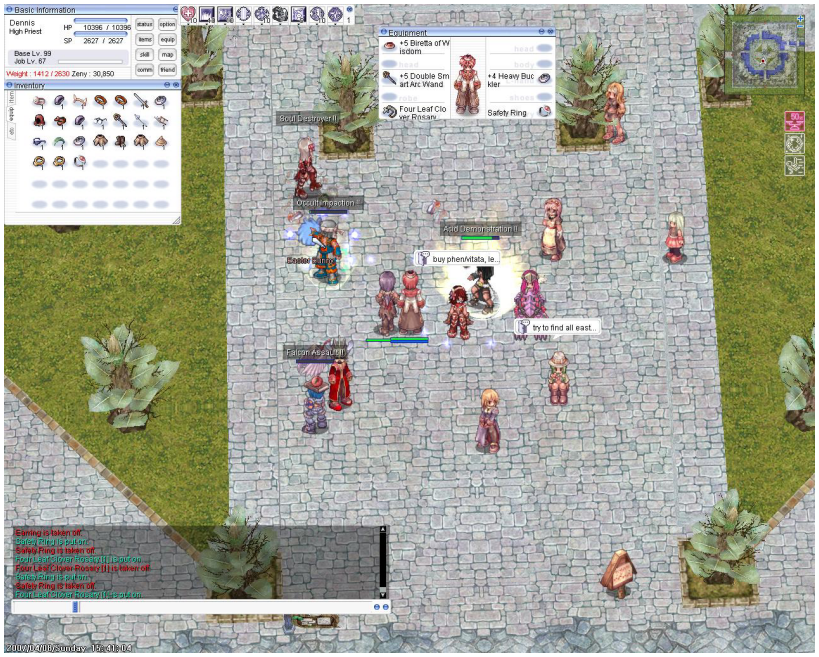


Figure 2: Screenshot from the game Ragnarok Online

client, as well as for bots driven by OpenKore. Within these packetstreams we were able to distinguish different kind of packets. Obviously we are in particular interested in packets which could help us in identifying bots and humans. Table 1 shows the packets we selected for our experiment. Along with the packet

MakeConnection	Starts a new session
MoveToXY	The player moves to a location
ChDir	Change viewing direction
TakeItem	The players picks up an item
Attack	The player attacks (a monster)

Table 1: Selected packets for investigation

information we stored a time stamp for each packet. Using these timestamps we calculated the following derived values for each session:

- *Number of packets per second.*
  All packets are counted (not only the selected ones), and divided by the duration of the session in seconds.
- *Average time between 'Move' packets.*
  The average time between two moves. In Ragnarok Online a player moves by selecting a target with the left mouse button.
- *Average distance between coordinates of 'Move' packet.*
  The Euclidean distances between the target of two subsequent moves (in game coordinates).
- *Average time between 'Take Item' packets.*
  The average time between two pickups. Most monsters drop items when killed.
- *Average time between 'Change Direction' and 'Take Item' packet.*
  If the character is not facing an object that he/she wants to pick up, it will change direction first. This feature measures the average time between the direction change, and the actual take item packet.

- *Average time between a kill and the first following 'Take Item' packet.*
  The average time it takes for the player to pick up an item after a monster is killed.

- *Number of attacks per second.*
  The number of attacks divided by the duration of the session in seconds.

These numerical values form a feature vector which (hopefully) represents the behaviour of a player.

Of course these features were selected carefully. Although all features showed tendencies which differ for bots and human players, none of them alone could determine if the session was human or bot driven. See for example Figure 3 which shows how in general bots have higher packet rates then human players, but the feature alone is not enough to decide who we are dealing with.

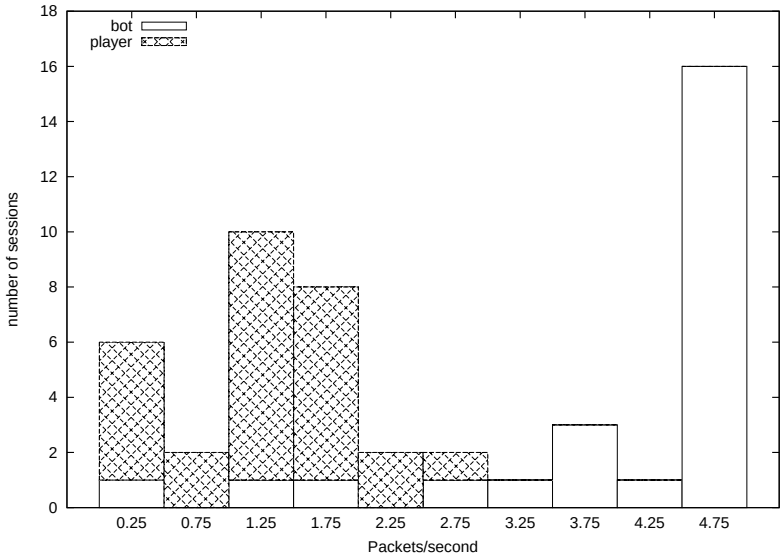


Figure 3: Histogram for number of packets/second feature

2.4 Classification

In order to classify our feature vectors we used a standard BackProp Neural Network with 7 input neurons (one for each feature), 7 hidden neurons and a single output neuron. Output activation '0' represents a player, while '1' designates a bot. We used the standard sigmoid threshold function, and learning factor $\eta = \frac{1}{2}$. We trained the network until the RMS error was below 0.0001.

To help out with the experiment we asked 25 players (of different jobs/levels) to play a session of at least 15 minutes. There were no restrictions on their behaviour whatsoever. Subsequently we configured 25 different OpenKore bots and observed them on different locations, fighting different monsters.

3 Results

Since the dataset is relatively small, we decided to create a trainingset of 49 sessions, and a testset of 1 single session. This procedure is repeated 50 times (leaving out every player/bot once). The results are presented in Table 2. The results show extremely good classification results, except for Player 13, Player 17, Bot 19 and Bot 22. Some further investigation learned that both player 13 and 17 used a special levelling technique (called mobbing) in which large groups of monsters are collected and killed together. It is possible that the character's movement during the collection phase has a mechanical touch and is misinterpreted by

test	activation	test	activation
Player 1	0.0004	Bot 1	0.9999
Player 2	0.0006	Bot 2	0.9997
Player 3	0.0005	Bot 3	0.9999
Player 4	0.0010	Bot 4	0.9999
Player 5	0.0003	Bot 5	0.9999
Player 6	0.0053	Bot 6	0.9999
Player 7	0.0003	Bot 7	0.9971
Player 8	0.0003	Bot 8	1.0000
Player 9	0.0002	Bot 9	0.9994
Player 10	0.0013	Bot 10	0.9999
Player 11	0.0004	Bot 11	0.9998
Player 12	0.0007	Bot 12	0.9999
Player 13	0.9975	Bot 13	1.0000
Player 14	0.0004	Bot 14	1.0000
Player 15	0.0007	Bot 15	0.9997
Player 16	0.0010	Bot 16	0.9999
Player 17	0.7023	Bot 17	0.9998
Player 18	0.0002	Bot 18	0.9999
Player 19	0.0007	Bot 19	0.0098
Player 20	0.0008	Bot 20	0.9999
Player 21	0.0008	Bot 21	0.9999
Player 22	0.0009	Bot 22	0.6731
Player 23	0.0001	Bot 23	0.9999
Player 24	0.0006	Bot 24	0.9999
Player 25	0.0007	Bot 25	0.9841

Table 2: Output activations for players/bots

the network. Both Bot 19 as 22 appeared to have rested long periods during their sessions. Since there was one player who spent her complete session sitting, it is possible that in those cases not enough feature information was present to enable proper determination.

All scores combined yield an average score of 94%. Thresholding at 0.5 would mean 4 errors in 50, so 92%.

4 Conclusions

Although the method presented in this paper is only validated in a small scale experiment, the results look promising. Knowing that (for now) only simple features were used and the session times were short, there appears to be enough room for improvements. A valuable feature might be the average angle between moves: bots often appear to make strange course corrections, while human players tend to move smoother, according to a plan.

Obviously, when bot developers find out what aspects of their bot makes it detectable by our method, they can adjust their bot to make it undetectable again. We do not consider this a weakness of our particular detection method. It is a mere fact that when a bot developer knows what it's bot is being tested on, they can make it undetectable again. Obviously, if a bot can mimic human actions perfectly, there is no way to distinguish a bot from a player by looking at it's behaviour.

Future research might consider the asymmetry in the decision making: having false positives (recognizing players as bots) is worse then having false negatives (bots are recognized as human players). For example, if the network's outcome is connected to an automatic jailing system[2] people might be rightfully upset if they are jailed without doing anything wrong.

[2]Ragnarok Online has a jail, in which players are placed if they violate the rules.

5 Acknowledgements

We like to thank Alchemist, Aligner, BamBam, buf priest, Cartimandua, dark acolyte, Datsylel, Ewan, Frea, Henkie, icarus, ilene, Kugutsu, maat van Ruben, Ophelia, Othello, Pebbles, Raio, Rhytor, Ruin, Rydia, Scarva, sniperboy, sniperboy's priest, and Xena for helping out with the experiments.

References

[1] N.E. Baughman and B.N. Levine. Cheat-proof playout for centralized and distributed online games. In *Proceedings of IEEE INFOCOM 2001*, pages 104–113, 2001.

[2] The OpenKore Project (Ragnarok Online Bot). `http://www.openkore.org`, last accessed August 2008.

[3] Y. Cai and P. Shackelford. *Networked Gaming: Driving the Future*. Parks Associates, Dalles, Texas USA, 2005.

[4] Kuan-Ta Chen, Jhih-Wei Jiang, Polly Huang, Hao-Hua Chu, Chin-Laung Lei, and Wen-Chin Chen. Identifying MMORPG bots: a traffic analysis approach. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, pages 4–12, New York, NY, USA, 2006. ACM.

[5] E. Cronin, B. Filstrup, and S. Jamin. Cheat-proofing dead reckoning multiplayer games. In *Proceedings of the* 2^{nd} *International Conference on Application and Development of Computer Games*, 2003.

[6] M. DeLap, B. Knutsson, H. Lu, O. Sokolsky, U. Sammapun, I. Lee, and C. Tsarouchis. Is runtime verification applicable to cheat detection? In *Proceedings of ACM SIGCOMM 2004 Workshops on NetGames '04*, pages 134–138, 2004.

[7] Philippe Golle and Nicolas Ducheneaut. Preventing bots from playing online games. *Comput. Entertain.*, 3(3):3–3, 2005.

[8] Ltd. Gravity Co. Ragnarok online. `http://www.ragnarokonline.com`, last accessed August 2008.

[9] B.S. Woodcock. An analysis of MMOG subscription growth. `http://www.mmogchart.com`, last accessed August 2008.

[10] Roman V. Yampolskiy and Venu Govindaraju. Embedded noninteractive continuous bot detection. *Computers in Entertainment*, 5(4):1–11, 2007.

[11] Jeff Yan and Brian Randell. A systematic classification of cheating in online games. In *NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–9, New York, NY, USA, 2005. ACM.

Hierarchical Planning and Learning for Automatic Solving of Sokoban Problems

Jean-Noël Demaret [a*] François Van Lishout [b] Pascal Gribomont [b]

[a] *Lab for User Cognition & Innovative Design, University of Liège, 4000 Liège, Belgium*
[b] *Montefiore Institute, University of Liège, 4000 Liège, Belgium*

Abstract

The most immediate method used to solve single-player games like Sokoban is to view them as state-space problems. However, classical search algorithms are insufficient to solve nontrivial problems and Sokoban remains a very challenging domain for the AI research community. The current state-of-the-art solver uses a heuristic-based approach coupled with a set of domain-specific enhancements. This paper introduces a new solving method which relies on two principles observable in human in-game behavior: (1) the decomposition of a complex problem into a sequence of simpler subproblems and (2) the non-repetition of encountered mistakes. We demonstrate the relevance of this approach and we show that it enables to reach performance comparable to the current state-of-the-art solver on a difficult 90-problem test suite.

1 Introduction

Artificial Intelligence researchers have always viewed games as privileged subjects of exploration to build computing machinery able to simulate human reasoning capabilities. In many games like Chess or Checkers, the machine has become stronger than the best human players. However, there are games where human performance remains significantly higher than those of the machine although their computing capabilities increase continuously. One of them is Sokoban.

1.1 Sokoban

Sokoban is a japanese single-player game where the player moves in a maze and has to push stones to specific locations, called *goals* (see Figure 1). The player can only push stones and he can only push one at a time. Solving a Sokoban problem consists in finding a sequence of moves that leads to a situation where all goals are filled, i.e., occupied by a stone. Most Sokoban problems admit several solutions. A solution can be expressed either in terms of player's moves or in terms of pushes. In this paper, we only consider the latter and an elementary game action will therefore not be a player's move but the push of a stone.

The most immediate method used to solve a Sokoban problem is to view it as a state-space problem. The structure of this state-space is a graph where each node is a game state whose successors are the game states reachable in one push. Solving a problem consists thus in finding a path in this graph between the node representing the initial game state and a node representing a solution game state, i.e., a game state where all goals are filled. However, classical search algorithms are insufficient to solve nontrivial problems. Sokoban has, in fact, several features that make it a particularly complex problem [4, 5, 9]. The branching factor of a problem, i.e., the number of possible pushes in a given game state, is very high (possibly more than 100) and a solution can be very long (more than 600 pushes). Thereby, the size of the state-space is enormous; it has been estimated at 10^{98} for a 20×20 maze. By comparison, in the case of Chess, the average branching factor is 35, the length of a game is about 50 moves and the size of the state-space is valued at about 10^{40}. Also, contrary to other games, a bad move can lead in Sokoban to a *deadlock*, a situation in which the solution

*F.R.S.-FNRS Research Fellow (Belgium)

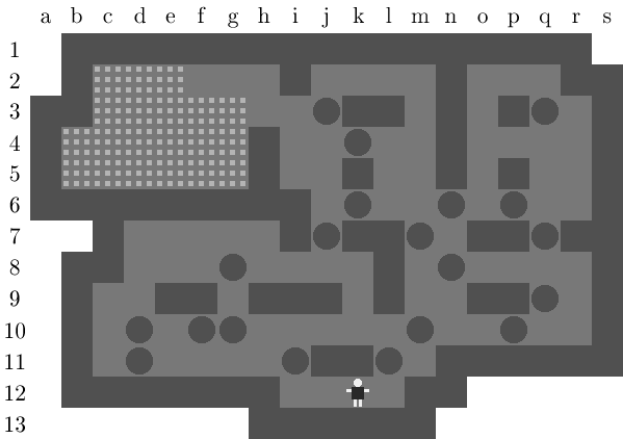


Figure 1: The problem #27 of the benchmark.

game state is not reachable anymore. Figure 3 shows such situation: there is no way to bring the three stones on *h8*, *h9* and *g10* to the goal aera. Finally, Sokoban's wide variety of problems makes it extremely difficult to find common strategies to solve them.

1.2 State of the Art

The program *Rolling Stone* [5] from the University of Alberta is currently the best documented Sokoban solver. It uses the heuristic search IDA*-algorithm [6] coupled with a set of domain-specific enhancements that enable it to solve 59 problems of a difficult 90-problem test suite[1]. Despite this significant result, it seems that the performance of this program, and therefore of this heuristic-based approach, has reached its limits. On the one hand, as explained in [5], a heuristic function is difficult to design and expensive to calculate ($O(n^3)$ for a problem of n stones). On the other hand, [5] also demonstrates that an informed search algorithm cannot solve any problem of the benchmark without the use of domain-specific enhancements. Finally, *Rolling Stone* no longer seems to be under research and recent publications on Sokoban from the University of Alberta explore different methods of resolution [2].

A completely different approach was adopted by *Talking Stones* [7], a program developed a the University of Liège. It relies on a multi-agent representation of a Sokoban problem which is to view the primitive elements of the game (the stones in the case of Sokoban) as agents which collaborate with each other to achieve a common objective. The program implements an algorithm capable of solving immediately the whole subclass of Sokoban problems satisfying the following three conditions:

- It must be possible to determine in advance the order in which goals will be filled.
- It must be possible to bring at least one stone in the first goal to be filled, without having to modify the position of another stone.
- For each stone satisfying the previous condition, the problem obtained by removing that stone and replacing the first goal by a wall must also belong to the subclass.

These conditions are very restrictive and most difficult Sokoban problems don't satisfy them (only one problem of the 90-problem benchmark). For this reason, *Talking Stones* uses a classical search algorithm to explore the state-space of the problem in order to find a game state belonging to the subclass. Note that it is trivially the case for the solution game state and that this solving protocol is therefore theoretically complete.The first version of the program is able to solve 9 problems of the benchmark without the use of any other enhancement. We will see that the method presented in this paper will allow us to eliminate the two last conditions and to achieve significantly better performance.

At present, no solver is able to solve all the problems of the benchmark and Sokoban still remains a very interesting domain for researchers working in the field of search and planning problems (e.g., [1]).

[1]This test suite is available at http://www.cs.cornell.edu/andru/xsokoban.html

2 Divide to Plan

The systematic exploration of all possibilities is an approach that suits the workings of a machine but does not correspond to the approach adopted by a human player facing a Sokoban problem. We can indeed notice that the human player builds his in-game behavior around the definition of high-level strategic objectives. An important point is that the player defines a strategic objective *before* searching a sequence of moves that would allow him to achieve it. This observation led us to address the automatic solving of Sokoban problems from a new angle, the hierarchical planning. The principle of this approach is to divide a complex problem into a sequence of subproblems of lesser importance in order to facilitate its resolution [8].

In the field of search problems, the use of hierarchical planning introduces an additional level of search involving actions of higher level than the elementary actions (e.g., moves) used by classical search algorithms to explore the state-space of a problem. In this approach, the problem is first solved using these high-level actions, then the realization of each high level action is transformed into a sequence of elementary actions to obtain the desired solution to the problem.

We have already applied this principle of hierarchical decomposition choosing to use the push of a stone as elementary game action. A push corresponds indeed to a sequence of player's moves and can therefore be regarded as a higher-level action.

2.1 Decomposition of a Sokoban Problem

How to divide a Sokoban problem into a sequence of subproblems? The decomposition that we have chosen is based on the following observation: the solution to a Sokoban problem can be divided into a series of sequences of pushes to the end of which a stone is brought into its final location, i.e., the goal it will occupy in the solution game state. Each of these sequences may itself be decomposed into two phases: (1) an *extrication phase*, during which a number of pushes are conducted on different stones, and (2) a *storage phase*, during which a succession of pushes relating to a single stone brings it into its final location. So, the planning phase of our solving method is to determine the order in which the goals will be filled. We call the produced ordered list of goals the *goal scheduling*. Each subproblem will therefore consist of finding a way to fill a given goal, i.e., a sequence of pushes such as we have described above.

It is interesting to notice that any scheduling will be a valid decomposition of a solution. Indeed the defined decomposition does not exclude that goals are filled during the extrication phase. Given a set of n goals to fill, in the worst case scenario, i.e., if the first goal of the scheduling is the last goal that has to be filled, the extrication phase of the first sequence will fill the $(n-1)$ first goals and the storage phase will bring the last stone into the last goal. In this scenario, the resolution of the first subproblem has led to solve the entire problem and the sequences of pushes corresponding to the resolution of the $(n-1)$ remaining subproblems will therefore be empty. However, such a situation loses the entire relevance of the method and is only useful to demonstrate its theoretical completeness.

Indeed, a scheduling will be meaningful only if it divides the solution in n non-empty sequences, i.e., if it really corresponds to the order in which the goals will effectively be filled. Actually, the main benefit of the method will be to significantly reduce the depth of the search tree associated to the resolution of each subproblem, replacing the storage phase of a sequence by a single game action that we will call a *macro-push*. For example, in Figure 3 (left), a macro-push could be used to bring the stone on *k9* to the goal *p4* in a single game action.

Our approach will thus be adapted to the subclass of Sokoban problems where it is possible to determine in advance and without ambiguity the order in which the goals will be filled. As we will see, this is mainly the case for problems having only one entrance and one goal area. An *entrance* is a location from which a stone can enter into a goal area. A *goal area* is a set of goals occupying contiguous locations.

2.2 Goal Scheduling

A goal scheduling is a permutation of the list of the n goals of a problem. The determination of a scheduling is to select a particular permutation among the $n!$ possible permutations of this list. The scheduling that we search is a scheduling which corresponds to the order in which the goals of the problem can effectively be filled. We call such a scheduling an *effective scheduling*. Formally, a scheduling will be effective for a given problem if at least one solution in which goals are filled in the order defined by this scheduling exists. This definition of effectiveness is however hardly exploitable. Indeed, proving that a scheduling is effective means finding a solution to the problem and therefore makes goal scheduling meaningless. It is thus essential

to identify a less restrictive condition able to generate a "good scheduling" from an operationally reasonable analysis of the initial game state of a problem.

A necessary condition for a scheduling to be effective is that the scheduling would not create any situation of deadlock, i.e., it does not lead to a situation in which the stones occupying filled goals make some unfilled goals unreachable. For example, in Figure 2, the goal *k7* cannot be filled before the goal *i7* without create a deadlock. We call *consistent scheduling* a scheduling which guarantees that no goal of the problem will ever be made unreachable by the stones occupying the filled goals. The latter is important because the consistency of a scheduling does not exclude that a goal is made unreachable by the other stones of the problem (see Figure 2).

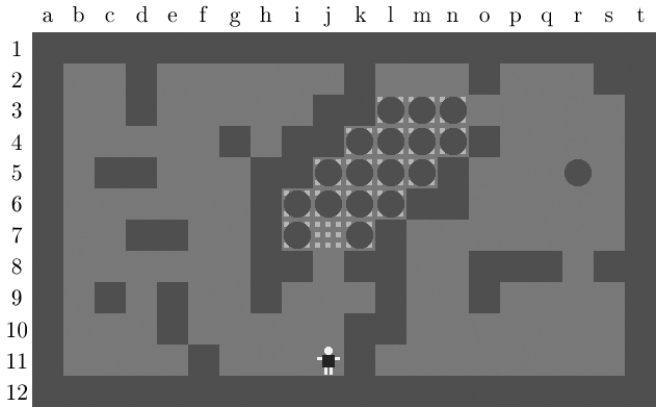


Figure 2: The goal *j7* could be theoretically filled through the entrance *j8* but the location of the stone *r4*, that cannot be bring to this entry, makes it effectively unreachable.

Therefore, we cannot exclude that a scheduling, though consistent, is such that all the sequences of pushes filling the goals in the order defined by the scheduling lead to a deadlock. A consistent scheduling is thus not always an effective one. Let $\mathcal{O}$ be the set of all $n!$ possible schedulings in a problem of n goals, $\mathcal{O}_{consistent}$, the set of all consistent schedulings and $\mathcal{O}_{effective}$, the set of all effective schedulings. We can establish the relationship

$$\mathcal{O}_{effective} \subseteq \mathcal{O}_{consistent} \subseteq \mathcal{O}$$

The proof that a scheduling is consistent is relatively easy. It consists of filling the goals of the problem one by one in the order defined by the scheduling and check at every step if all unfilled goals are reachable. So, the consistency will be the criterion we will use to generate a "good scheduling". This solution is not ideal since it does not guarantee that the produced scheduling is always effective. However, the condition of consistency is sufficient for a subclass of Sokoban problems defined by an easily identifiable feature.

Indeed, in a problem having only one entrance (and therefore, one goal area), the accessibility of a goal only depends on the occupation of the goal area and not on the possibility to bring a stone to a given entry. We have therefore for this situation the relationship

$$\mathcal{O}_{effective} = \mathcal{O}_{consistent}$$

In this case, the scheduling of goals can be determined without ambiguity since the generated consistent scheduling will always be effective and not arbitrarily chosen in the set of all consistent schedulings.

Our goal scheduling algorithm starts from a game state in which the goal area is fully occupied by stones. It consists then in emptying it by gradually evacuating the stones one by one either through the single entry or through an arbitrarily chosen one. The scheduling of the goals is then obtained by reversing the order in which the goals have been emptied.

3 Learn from Deadlocks

An important feature of an intelligent agent is its ability to learn. In its simplest form, it is reflected by its capability to not commit the same mistake twice. In the case of solving a Sokoban problem, making a mistake is to create a deadlock. In most cases, a deadlock situation is not induced by the position of all the stones of the problem but by a subset of them. Accordingly, all game states that contain this subset, which we call a *local deadlock*, are also deadlock situations (see Figure 3). There may be a potentially very high number of such game states. This number is usually inversely proportional to the number of stones in the subset. Identifying and remembering this subset of stones makes it possible to exclude from the search tree a significant number of game states and thus reduce significantly its branching factor.

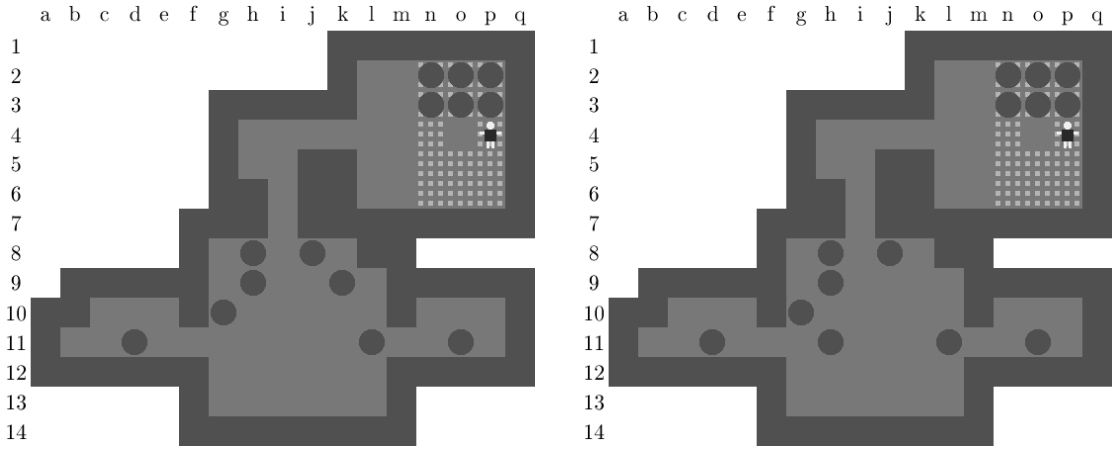


Figure 3: Two game states that contain the same local deadlock (*h8*,*h9*,*g10*).

It is important to notice that some deadlock situations also depend on the player's position. For example, the game states showed in Figure 3 would not be deadlock stiuations if the player was on *g8*. That information will therefore be associated with each stored subset of stones so as not to exclude game states from the search tree that would be wrongfully identified as deadlock situations.

In order to avoid using too many resources, the identification of local deadlocks is limited to the analysis of game states from which a subproblem has not been resolved. It is, indeed, reasonable to assume that such a game state will contain most of the time at least a subset of stones involving a deadlock.

Also, the number of existing subsets of stones increases exponentially with the number of stones present in the analyzed game state. It is thus not feasible to examine each subset of stones thoroughly. It is therefore necessary to choose a limited number of relevant subsets of stones. In our experiments, we have noticed that many local deadlocks were composed of groups of contiguous stones. We have therefore chosen to use these groups of contiguous stones, which we call clusters, as constituents for examined subsets. The examined subsets will be the subsets consisting of one, two or three clusters.

The method used to determine if a given subset of stones implies a deadlock uses a classical search algorithm. It searches a way to fill the target goal of the subproblem for which no solution has been found, from a temporary game state in which stones outside the subset were removed. If it is not possible to bring a stone towards the target goal, the subset is identified as a local deadlock and is added to the database.

4 Solving Protocol

Conceptually, our solving protocol is organized around four search functions which are each responsible for solving a particular problem :

- The first function is in charge of determining the goal scheduling by analyzing the initial game state.
- The task of the second function is to fill one after the other the goals of the problem in the order defined by the scheduling.

- The third function is in charge of solving the subproblem consisting of filling the current target goal, i.e., the first unfilled goal of the scheduling, from a given game state.
- The role of the fourth function is to search for possible local deadlocks in a given game state.

Starting from these four search functions, our solving method can be described as follows. To begin with, the first function analyzes the initial game state and determines the order in which the goals are to be filled. The resulting goal scheduling and the initial game state are then passed as arguments to the second function. The latter then searches a way of filling the first goal of the scheduling. To do so, it gives control to the third function and passes it the initial game state and the position of the current target goal as arguments. The third function launches a search which aims at finding a sequence of moves leading to a game state in which the target goal is filled. This research is implemented as a classical search algorithm which used a transposition table in order to avoid visiting the same game state twice and verifies at each step the possibility of performing a macro-push to the target goal. The solution node found is returned to the second function which uses the game state content in this node as a starting point from which a way of filling the second goal of the scheduling will be found. If, at any step of the solving process, a way of filling the nth goal cannot be found, a search for local deadlocks is made (fourth function) and another way of filling the $(n-1)$th goal is then searched.

Operationally, the second function can be seen as doing a depth-first walk in a tree whose root is the initial game state of the problem and where a node located at a depth n is a game state in which the $(n-1)$th first goals of the scheduling are filled. In this tree, a node at depth n has, as successors, the game states reachable in any number of moves and where the first n goals of the scheduling are filled.

4.1 Results

In order to demonstrate the relevance of our approach with experimental results, we have implemented the introduced solving protocol as a Scheme program. A detailed description of this implementation is given in [3]. We have then evaluated its performance on the classical 90-problem test suite used by the previously mentioned start-of-the-art programs. Our program was able to solve 54 problems of this difficult benchmark.

During our experimentations, we realized that some problems for which no consistent goal scheduling could be found, became accessible and sometimes easily solvable by our program if the disposition of the goals was previously rearranged. In fact, to solve those problems, it is necessary to bring stones to temporary positions before bringing them to their final location in order to avoid making some of the goals unreachable. Figure 4 shows an example of such rearrangement.

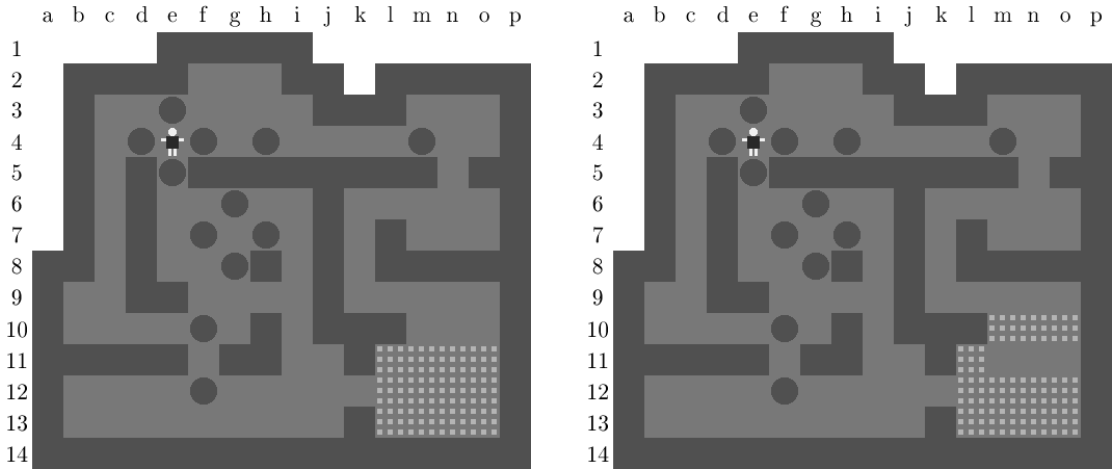


Figure 4: An example of goal rearrangement (right) for the problem #87 of the benchmark (left).

It is easy to be convinced that obtaining a solution to the initial problem from the one of the rearranged problem is almost immediate. Therefore, the design of a procedure to automate this rearrangement would allow us to add 7 problems to the list of problems solved by our program. It would therefore be able to solve

61 of the 90 problems of the benchmark and thus to compete with the performance of the state-of-the-art solver *Rolling Stone* (59 problems solved).

Table 1 shows results obtained on the 61 solved problems. Total nodes represents the total number of nodes that were explored during the resolution of subproblems and the search for local deadlocks. The next column shows the total nodes obtained by *Rolling Stone* [5]. At last, Time is the time consumed for solving the problem on a standard desktop machine[2].

Problem	Total nodes	Rolling Stone total nodes	Time	Problem	Total nodes	Rolling Stone total nodes	Time
1	97	1,267	1 s	51	121	29,569	6 s
2	291	7,530	3 s	53	15	22,308	8 s
3	32	14,095	1 s	54	162	66,306	19 s
4	6,858	50,369	1 min 26 s	55	15	2,993	6 s
5	133	35,974	4 s	56	35	50,924	7 s
6	193	5,503	2 s	57	65	128,282	21 s
7	427	15,790	4 s	58	306	138,838	10 s
8	77	409,714	14 s	59	272	409,470	13 s
9	2,193	407,103	27 s	60	182	31,413	8 s
10	710	19,967,875	3 min 51 s	61	310	77,555	21 s
11	83,711	2,331,950	27 min 25 s	62	1,649	69,728	56 s
12	31	372,264	10 s	63	40	578,066	10 s
13	17,918	unsolved	11 min 2 s	64	154	186,508	8 s
17	1,427	33,901	13 s	65	122	23,004	10 s
19	454	6,089,182	20 s	67	580	104,356	25 s
21	2,402	258,852	36 s	68	395	236,157	13 s
24	403,229	unsolved	3 h 2 min	70	2,161	178,657	1 min 34 s
25	516,000	592,585	6 h 11 min	72	86	45,735	9 s
26	1,171	126,379	19 s	73	51	103,494	9 s
27	10,183	unsolved	1 min 46 s	75	2,133,007	5,095,054	6 h 23 min
34	503	442,025	12 s	76	90,592	1,980,094	27 min 27 s
35*	168	unsolved	21 s	78	8	4,913	2 s
36	147,434	5,785,290	1 h 5 min	79	204	13,114	11 s
37	1,078	unsolved	1 min 5 s	80	344	26,309	10 s
38	14,749	56,563	2 min 9 s	81	2,604	206,423	40 s
39*	1,646,578	unsolved	5 h 50 min	82	66	45,014	4 s
43	1166	523,907	16 s	83	1040	6,856	19 s
44*	736,375	unsolved	2 h 14 min	84	695	7,818	52 s
45	942,964	410,134	3 h 13 min	86*	143	unsolved	2 s
46*	53,584	unsolved	14 min 31 s	87*	80	unsolved	4 s
47*	264	unsolved	5 s				

Table 1: Results obtained on the 61 solved problems. Rearranged problems are marked with an asterisk (*).

4.2 Comparison with Other Methods

This work is the continuation of research conducted at the University of Liège in 2005 around the solver *Talking Stones*. Note that the subclass of problems that could be solved efficiently by our method includes all the problems solved immediately by *Talking Stones*. Indeed, these problems should satisfy three conditions, the first one being that the order in which the goals will be filled can be determined in advance. From this point of view, our method can be regarded as a generalization of the solving protocol used by *Talking Stones* since the necessity to satisfy the two other very restrictive conditions has been eliminated.

Similarly, we can establish links between the ideas behind our approach and some of the domain-specific enhancements used by *Rolling Stone*. First, the use of a macro-push to replace the storage phase in the resolution of a subproblem is similar to *Goal Macros* (see [5], enhancement #5), which consists in bringing a stone entering in a goal area immediately into an appropriate final location. Secondly, *Pattern Search* (see [5], enhancement #7) is used in *Rolling Stone* to analyze some game states deemed critical (by a specific heuristic) to identify potential conflicts existing within subsets of stones. Those subsets are stored in memory and then used to adjust the estimation produced by the heuristic function. The idea behind this improvement is therefore similar to the one consisting of searching for local deadlocks. Finally, like *Rolling Stone*, we use a transposition table (see [5], enhancement #1) to avoid visiting the same node of the search tree twice during the resolution of a subproblem.

In [5], the authors show that the three enhancements described above are the most important in term of performance for *Rolling Stone*. This assertion is clearly confirmed by our approach which is able to reach comparable results with a solving protocol which is mainly based on ideas similar to those that support them. Furthermore, [5] also demonstrates that the use of a heuristic function is insufficient but necessary to

[2] 1.8GHz CPU and 512 MB RAM

solve any problem of the benchmark. Our results suggest instead that the relevance of a heuristic function in comparison with the domain-specific enhancements must be relativized in the case of Sokoban.

5 Conclusion and Future Work

This paper focuses on automatic solving of Sokoban problems. From the observation of human in-game behavior, we have developed a new solving method based on planning by decomposition into subproblems. We have described such decomposition and we have demonstrated its relevance for the subclass of Sokoban problems where it is possible to determine in advance the order in which the goals will be filled. We have then incorporated in our method a fundamental learning concept. By analyzing encountered deadlocks and by storing collected information in a database, our program was able to eliminate many game states from the search tree. The solver implemented from the resulting solving protocol was thus able to solve 54 problems of a difficult 90-problem test suite. At last, we have shown that a preliminary goal rearrangement allows our program to solve 61 problems of the benchmark and therefore to reach performance comparable to the currently best documented solver (59 problems solved).

The major limitation of the introduced method resides in its inability to efficiently handle the existence of several entrances and consequently the existence of several goal areas in a problem. In this type of problem, a preliminary analysis of the initial game situation is not sufficient to determine an effective goal scheduling. To overcome this limitation, we consider exploring the possibility of not determining in advance a full goal scheduling but recomputing a list of possible target goals at each step of the solving process.

In future work, we plan also to develop an automatic goal rearrangement procedure. This procedure will take the form of an additional search function which will be called when no goal scheduling can be found. A valid goal rearrangement must have two properties: (1) it must be possible to find a goal scheduling for the rearranged problem and (2) it must be possible to reach a game situation in which the initial problem is solved from the solved rearranged problem.

Finally, we plan to inject in our program the heuristic function and the domain-specific enhancements introduced in [5] to improve the search algorithm used in subproblem resolution. We hope that the combination of a high-level planning and learning approach with these lower-level search optimizations will lead to significant advancement in the challenging domain of automatic solving of Sokoban problems.

References

[1] M. S. Berger and J. H. Lawton. Multi-agent planning in Sokoban. *Lecture Notes in Computer Science*, 4696:334–336, 2007.

[2] A. Botea, M. Muller, and J. Schaeffer. Using abstraction for planning in Sokoban. *Lecture Notes in Computer Science*, 2883:360–375, 2003.

[3] J.-N. Demaret. *L'intelligence artificielle et les jeux : cas du Sokoban*. Master thesis, University of Liège, 2007.

[4] A. Junghanns. *Pushing the limits: New developments in single-agent search*. PhD thesis, University of Alberta, 1999.

[5] A. Junghanns and J. Schaeffer. Sokoban: Enhancing general single-agent search methods using domain knowledge. *Artificial Intelligence*, 129(1):219–251, 2001.

[6] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 26(1):35–77, 1985.

[7] F. Van Lishout and P. Gribomont. Single-player games: Introduction to a new solving method combining state-space modelling with a multi-agent representation. In *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'2006)*, pages 331–337, 2006.

[8] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 1995.

[9] W. Wesselink and H. Zantema. Shortest solutions for Sokoban. In *Proceedings of the 15th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'2003)*, pages 323–330, 2003.

Mixed-Integer Bayesian Optimization Utilizing A-Priori Knowledge on Parameter Dependences

Michael T. M. Emmerich [a] Rui Li [a] Anyi Zhang [a] Ildikó Flesch[b]
Peter Lucas[c]

[a] *LIACS, Leiden University, Niels Bohrweg 1, 2333-CA Leiden, The Netherlands*
[b] *Tilburg Center for Creative Computing, Tilburg University, Warandelaan 2, 5037 AB, Tilburg, The Netherlands*
[c] *Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands*

Abstract

Mixed-integer optimization problems arise in various application fields, such as chemical engineering and the optimization of medical image processing pipelines. Stochastic optimization algorithms, such as evolution strategies and estimation of distribution algorithms, can be used as solution methods for solving these problems approximately. Especially for real-world problems they often prove to be powerful methods due to their flexibility and robustness.

This paper introduces a new estimation of distribution algorithm that extends the Bayesian optimization algorithm (with fixed network structure) from binary optimization problems to mixed-integer optimization problems. We show that a-priori knowledge on dependences between decision variables can be exploited by this algorithm in order to improve convergence speed and reliability. In discussing the properties of heterogeneous Bayesian networks, representing multivariate distributions of mixed-variable type, we point out which kind of dependence information can be utilized. We assess the performance of the new approach using mixed-integer Nk-landscape models.

1 Introduction

Mixed-integer optimization deals with objective functions, the decision variables of which are of mixed type, for instance continuous, nominal discrete, and ordinal discrete. Mixed-integer optimization problems arise in various application fields, such as chemical engineering [6], optical filter design [15], or machine learning [11]. Stochastic optimization algorithms, such as Mixed-Integer Evolution Strategies (MIES) [6], proved to be powerful and flexible metaheuristics for solving real-world mixed-integer optimization problems.

A shortcoming of existing mixed-integer evolutionary algorithms, such as MIES, is that their variation procedures mutate each decision variable independently. Therefore, dependences between variables, even if they are known a-priori, cannot be taken into account. This contribution aims at designing and testing a mixed integer evolutionary algorithm that can utilize knowledge about such dependences.

One of the new types of evolutionary algorithms are the so-called Estimation of Distribution Algorithms (EDAs). EDAs do neither have a crossover nor a mutation operator. Instead, a new population is generated by sampling the probability distribution, which is estimated and updated based on the distribution of recently obtained 'successful' individuals. Different instantiations of EDA differ by the distribution types and update rules they use. For instance, the classical population-based incremental learning (PBIL) algorithm samples from an independent joint distribution of Bernoulli type [2], while the Univariate Marginal Distribution Algorithm (UMDA) [8, 17] features independent joint distributions of Gaussian type.

In this paper, we propose a Mixed-Integer Bayesian Optimization Algorithm (MIBOA), that is a variant of EDAs. MIBOA is an extension of the BOA approach to mixed-integer spaces using special types of Bayesian networks dealing with random variables of mixed-type. Moreover, a special type of mixed-integer Nk-landscape [1, 10] will be introduced that is well suited for testing the new approach.

The development of the new approach is motivated by problems in medical image analysis where the parameters of a medical image processing pipeline are to be optimized. Though the optimization of these systems is essentialy a black-box optimization problem, dependence information can be extracted heuristically from the known structure of the processing pipeline. For details on the application domain the interested reader is referred to [5].

The paper is structured as follows: Section 2 introduces the problem definition. In Section 3 we discuss briefly mixed-integer evolution strategies (MIES) and estimation of distribution algorithms with independent sampling distributions. Section 4 introduces Bayesian optimization and generalizes it to the mixed-integer case. After introducing test problems based on Nk-landscapes in Section 5, we present results of mixed-integer BOA on these landscapes. Finally, the main results of the paper are summarized and directions of future research are discussed.

2 Problem Definition of Mixed-Integer Optimization

In this contribution we define the mixed-integer optimization as follows:

$$\text{minimize } f(\mathbf{r},\mathbf{z},\mathbf{d}), \mathbf{r} \in \mathbb{R}^l, \mathbf{z} \in \mathbb{Z}^m, \mathbf{d} \in D_1 \times \dots D_n \tag{1}$$

Here, $\mathbf{r}$ denotes a vector of real numbers, $\mathbf{z}$ from a finite set of integer values (or ordinal discrete values), whereas $\mathbf{d}$ defines a n-tuple of nominal discrete variables with finite domains $D_i, i = 1,\dots,n$. The function f is considered to be a black-box function, or more precisely a function the mathematical structure of which is mainly unknown to the user. The only a-priori knowledge that we can exploit about f are assumptions about parameter dependences (interaction of variables). A common feature of functions in which interactions occur is that they cannot be decomposed into a sum of functions depending only on single variables. For example, if r_1 interacts with z_1 and all other parameters are independent from each other, we can write the function as:

$$f(\mathbf{r},\mathbf{z},\mathbf{d}) \equiv f_{1,l+1}(r_1,z_1)+f_2(r_2)+\dots+f_l(r_l)+f_{l+2}(z_2)\dots f_{l+m}(z_m)+f_{l+m+1}(d_1)+\dots+f_{l+m+n}(d_n)$$

and $f_{1,l+1}(r_1,z_1)$ cannot be written as a sum of functions of r_1 and z_1. Non-separability makes it potentially difficult to optimize these functions by optimization routines that exploit such an assumption, such as coordinate search but also evolutionary algorithms that mutate variable independently from each other. In Section 5, with the ADG-based Nk-landscapes, an example for a function class in which various variable interactions can be introduced will be discussed.

3 Algorithms with independent sampling distributions

Next, let us introduce the *evolution strategy* (ES) and the *estimation of distribution algorithm* (EDA) as two basic evolutionary algorithms for parameter optimization[1]: The *canonical* $(\mu+\lambda)$ *evolution strategy* has the following iteration scheme:

Step 1 : Create initial population $P \leftarrow \{(a_1,\varsigma_1),\dots,(a_\mu.\varsigma_\mu)\}$, where ς_i denotes a vector of dispersion parameters of the mutation distribution, e.g. standard deviations or mutation probabilities.

Step 2 : Create offspring population Q of size λ by choosing randomly elements from P and mutating first the distribution parameters ς_i to ς_i' and then the object variables a_i using distribution parameters ς_i'.

Step 3 : Set P to the μ best points (with respect to f) coupled with their mutated distribution parameters ς' out of $P \cup Q$.

Step 4 : If termination criterion reached return best found solution, otherwise go to Step 2.

In contrast to this, *estimation of distribution algorithms* apply the following main loop:

Step 1 : Initialize distribution parameters of distribution $\mathcal{D}_\theta$.

Step 2 : Create offspring population Q of size λ by sampling from the distribution $\mathcal{D}_\theta$.

[1] The ES is introduced, as it is a state-of-the- art technique in mixed integer optimization we will compare to later.

Step 3 : Set P to the μ best points in Q with respect to f.

Step 4 : Update parameters θ of the distribution $\mathcal{D}_\theta$ as a weighted average of the estimation of θ based on P and the current parameter set θ.

Step 5 : If termination criterion reached then return best found solution, otherwise go to Step 2.

While in ES the basic variation operator is *mutation*, the variation operator in EDA is *sampling* from a multivariate distribution the parameters of which are dynamically updated based on positive examples.

Next, let us describe the mutation and sampling procedure for the mixed-integer case (without parameter dependences).

The mutation of mixed-integer evolution strategies can be described as a procedure:

Continuous mutation: Set $r_i = r_i + \texttt{Normal}(0, s_r)$, $i = 1, \ldots, l$.

Integer mutation: Set $z_i = z_i + \texttt{Geometric}(0, s_z) - \texttt{Geometric}(0, s_z)$, $i = 1, \ldots, l$.

Nominal discrete mutation: If $\texttt{Uniform}(0, 1) < p_d$ set d_i to random value from $D_i - \{d_i\}$.

Here $\texttt{Normal}(0, s_r)$ computes a normally distributed random number with standard deviation parameter s_r, $\texttt{Geometric}(0, s_z)$ generates geometrically distributed random variables with mean s_z [11], while $\texttt{Uniform}(0, 1)$ generates a uniformly distributed random number between 0 and 1. Before the mutation of the distribution parameter s_r we employ the log-normal distribution as proposed by Schwefel [16] et al. $s_r \leftarrow s_r \exp(\tau_r \texttt{Normal}(0, 1))$ with $\tau_r = 1/\sqrt{l}$ being the learning rate. Accordingly, $s_z \leftarrow s_z \exp(\tau_z \texttt{Normal}(0, 1))$, with $\tau_z = 1/\sqrt{m}$ is used to adapt the step-size for integer mutations. The probability parameter p_d is mutated based on a logistic mutation see (e.g. [15] et al.) that makes sure that the value of p_d stays in $]0, 1[$. All three mutations of strategy parameters have the property, that increments are as likely as decrements of the value. The ES discussed here is termed mixed-integer evolution strategy and was discussed in several publications [6, 11].

For the *sampling* in the mixed-integer estimation of distribution algorithm similar distribution types are used. We employ the joint distribution $\mathcal{D}_\theta$ composed of

- a vector of l independent multivariate normal distributions, with mean values $\mu_1, \ldots, \mu_l$ and standard deviations $\sigma_1, \ldots, \sigma_l$.
- a vector of m random variables of type $\xi_i + Z_1(s_z) - Z_2(s_z)$, whereas $Z_1(s_z)$ and $Z_2(s_z)$ denote indentically independent geometrically distributed random variables with mean value s_z.
- a vector of n Bernoulli distributed binary random variables with probability parameters $p_1, \ldots, p_n$.

The described estimation of distribution algorithm is new for the mixed-integer search space. However, for binary nominal discrete parameters the algorithm is the classical population based incremental learning (PBIL) algorithm [2] and reduced to its continuous part it equals the so-called Univariate Marginal Distribution Algorithm (UMDA) [17, 8]. In the sequel, we will refer to the EDA algorithm for mixed-integer search space as MIPBIL.

The aforementioned two algorithms are used as reference algorithms to find out whether the introduction of dependence information improves the algorithms behavior or not. Next, we will look at an extension of MIPBIL that allows to integrate dependence information.

4 Mixed-Integer Bayesian Optimization Algorithm

In order to design a new mixed-integer estimation of distribution algorithm that can take into account dependences between variables of the objective functions we will replace the independent joint distribution $\mathcal{D}_\theta$ used in the MIPBIL approach by an heterogeneous Bayesian network with fixed structure. This approach is also used in the bayesian optimization algorithm (BOA) by Pelikan et al. [14]. Their BOA method is applied for binary search spaces and also learns the structure of the network, while our approach is defined for mixed-integer search spaces and requires a-priori knowlege on the dependency structure of variables in the objective function. To emphasize the similarity to the BOA algorithm, we will term the new approach Mixed-Integer BOA (MIBOA).

Bayesian networks yield very powerful probabilistic graphical representations. The key to their popularity is their ease of representation of independence relations, and their support for reasoning with uncertainty.

A Bayesian network is a graphical representation of a probabilistic problem, formally defined as a pair $\mathcal{B} = (G, P)$, where P is the joint probability distribution on the set of random variables and G is an ADG representing the dependence and independence relations among this set of random variables, where each graphically represented marginal and conditional independence also has to be valid in the joint probability distribution [13]. Clearly, the definition of Bayesian networks implies as well that a dependence in the graph does not have to define a dependence in the joint probability distribution P.

Let $\{X_1, \ldots, X_d\}$ be a set of random variables. Then, based on the independence relations in the graph G, the joint probability distribution P can be factorised as follows:

$$P(X_1, \ldots, X_d) = \prod_{v=1}^{d} P(X_v \mid \pi(X_v)), \tag{2}$$

where $\pi(X_v)$ denotes the graphically represented set of parents of random variable X_v. This implies that a joint probability distribution can be defined in terms of local distributions resulting in significant computational savings.

For reasoning in Bayesian networks there are several exact methods proposed that make use of local computations [4]. Here, local computations are based on the construction of join trees.

Hybrid Bayesian networks consist of both discrete and continuous random variables [3]. In these networks, local computations are possible, however, the correctness of the inference method depends on whether parents of a variable are discrete, continuous, a mixture of discrete and continuous, and on the choice of the local probability distribution.

The first method, introduced by Lauritzen [9], using *exact* inference is based on conditional Gaussian distributions. The restriction of this inference is that discrete random variables are not allowed to have continuous parents when hybrid Bayesian networks are concerned. To overcome this problem, Koller proposed a method which defines the distribution of these discrete nodes by a mixture of exponentials. However, for the inference Monte Carlo methods are used [7]. As another solution to this problem, we may discretise continuous variables, but discretisation introduces errors because we use approximation methods. However, in the experiment performed in this contribution we did not yet study the case of discrete nodes having continuous parents. For the Bayesian networks related experiments the BNT tool developed by Murphy was used [12]. The same basic algorithm than for PBIL was used, except that the distribution type and the update procedure was changed. A detailed description of the update algorithm would exceed the scope of the paper, and we refer to [12].

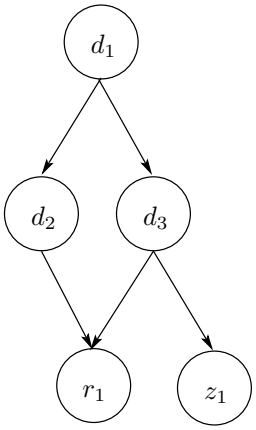
5 ADG-based Nk-landscapes

ADG-based Nk-landscapes (ADG-NKL), that we will introduce next, are attractive as models for optimization as their interaction structure corresponds to the dependence structure of Bayesian networks. Let $x_1, \ldots, x_d$ denote a set of decision variables (the type of which can be continuous or discrete) and assume the interaction structure of the function is described by some ADGs, which is basically defined by a function $\pi(\cdot)$, that assigns the set of parent nodes to each node, where the nodes represent parameters to be optimized. Then the ADG-based Nk-landscape can be written as a function of component functions f_i:

$$f(x_1, \ldots, x_d) = \sum_{i=1}^{d} f_i(x_i, \pi(x_i)) \tag{3}$$

Note, that this expression has the same structure as the expression $\log P(X_1, \ldots, X_d)$ (see Equation (2)). Note also, that the x_1, ..., x_d denote variables of the objective function in contrast to $X_1, \ldots, X_d$ denoting random variables.

The construction of the ADG-based Nk-landscapes corresponds to that of classical mixed-integer Nk-landscapes [10] with one exception. As for classical Nk-landscapes for each decision variable (or *gene*) x_i we choose k epistatic genes that interact with x_i, in ADG-based Nk-landscapes we chose exactly the parent nodes as epistatic genes. Note, that the number of them can vary with the index of the decision variable in question. That is why the k in the expression 'Nk-landscape' is not referring to the number of epistatic genes anymore - we kept it, however in the term, as it makes it easier to match the corresponding well known



d1	f1
0	0.12
1	0.45

d3	d1	f3
0	0	0.42
0	1	0.56
1	0	0.67
1	1	0.92

r1	d2	d3	f4
0	0	0	0.34
0	0	1	0.26
0	1	0	0.45
0	1	1	0.21
1	0	0	0.81
1	0	1	0.46
1	1	0	0.51
1	1	1	0.48

d2	d1	f2
0	0	0.32
0	1	0.43
1	0	0.73
1	1	0.13

z1	d3	f5
0	0	0.98
0	1	0.34
1	0	0.84
1	1	0.09

$$f(x) = f_1(d_1) + f_2(d_2, d_1) + f_3(d_3, d_1) + f_4(r_1, d_1, d_3) + f_5(z_1, d_3)$$

Figure 1: Example for an ADG-based Nk-landscape. The function values at the edge of the search space $[0, 1]^d$ are set randomly between 0 and 1. Values inbetween are interpolated [10].

Nk-landscapes with the ADG-based Nk-landscapes. As with classical Nk-landscapes, the definition of the component functions in ADG-based Nk-landscapes is based on randomly generated function tables [10], as visualized in Figure 1. In the mixed-integer case multilinear functions are used to interpolate between the randomly chosen function values at the edges of a hypercube as described in [10].

6 Results

In order to check whether a-priori knowledge on the interaction structure integrated in the structure of the Bayesian network helps to speed up search we have conducted experiments on various ADG types that are visualized in Figure 2. These ADGs were used to construct Nk-landscapes indicating that the represented independence and dependence relations in an ADG are also included in the Nk-landscape constructed from this ADG. The same ADG is used as a structure for the Bayesian network as a-priori knowledge. For the probability tables, however, no a-priori knowledge is used. They are initialized based on the first population of selected individuals.

We applied three types of algorithms on ADG-based Nk-landscapes. 15 variables are considered, 5 for each type (l=m=n=5). As the population size turned out to be a crucial parameter, two different population sizes, 28 and 100, are tried. A number of 20 runs were statistically evaluated for each strategy.

Figures 3 to 5 show convergence dynamics for different sample landscapes defined by their ADG, each of which with a different structure. Averaged objective function values (difference to the global optimum) and standard deviations are plotted over the number of evaluations performed.

On the landscape 'chain' (Figure 3), the MIBOA performs best, when the population size is set to 100. For a population size of 28 the MIBOA performs almost equally to the MIES. In both cases the MIPBIL algorithm was clearly outperformed.

On the landscape 'bitree' (Figure 4), a binary tree, the MIBOA performs best, when the population size is set to 100. For a population size of 28 the MIBOA is faster but in the long run MIPBIL results in (almost) the same good value. MIES seems to have a problem in this landscape, which may be due to step-size reduction which can be harmful in multimodal landscapes. The large standard deviation supports this conjecture.

On the landscape 'invtree' (Figure 5), again the MIBOA has a big advantage in the beginning. Here this acceleration is more visible than for the previous landscape types. Again the MIES algorithm seems to have problems to converge to the global optimum, while the MIPBIL is more reliable, but suffers from a low convergence speed.

Comparing a population size of 100 with a population size of 28, it was observed that the MIBOA algorithm performs better with the larger population size. The standard deviation of results in that case is remarkably lower, indicating a good reliability of the good results. In Table 1 we summarize more results, including the ADG types 'tritree', 'struct2', and 'struct3'. The ranking after 2000, 5000, 10000, and 20000

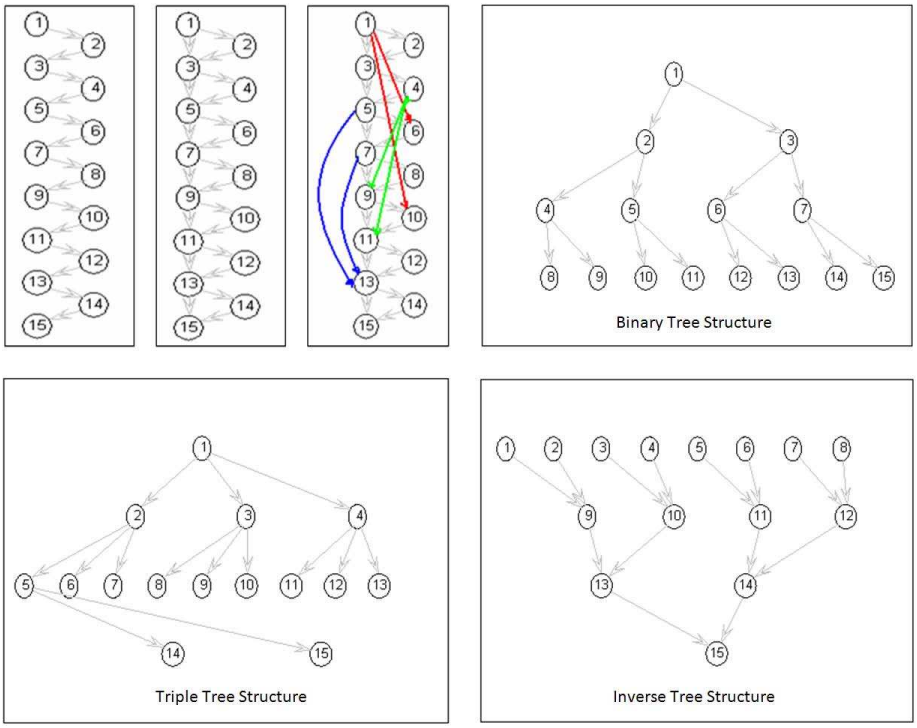


Figure 2: Various types of ADGs used to define ADG-based Nk-landscapes and corresponding Bayesian networks. From left to right, ADGs are termed 'chain', 'struct2', 'struct3', 'bitree', 'tritree', and 'invtree'. Node types are defined as follows: discrete nodes(1-5), continuous nodes(6-10), integer nodes(11-15).

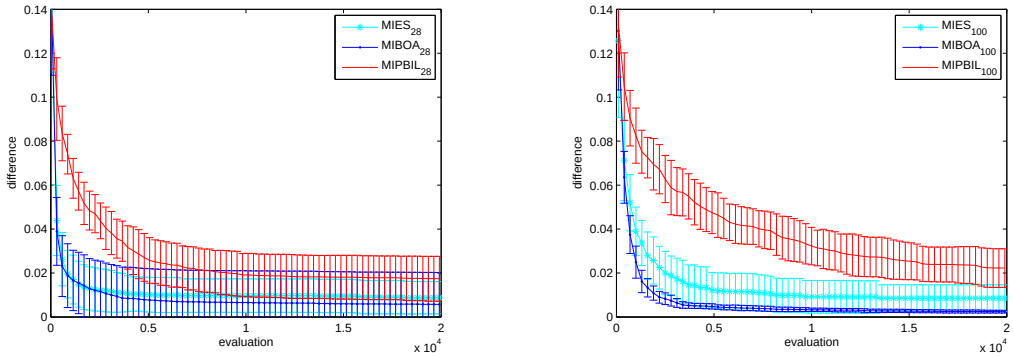


Figure 3: Convergence dynamics of MIES, MIPBIL, and MIBOA on a 'chain'-type ADG-NKL.

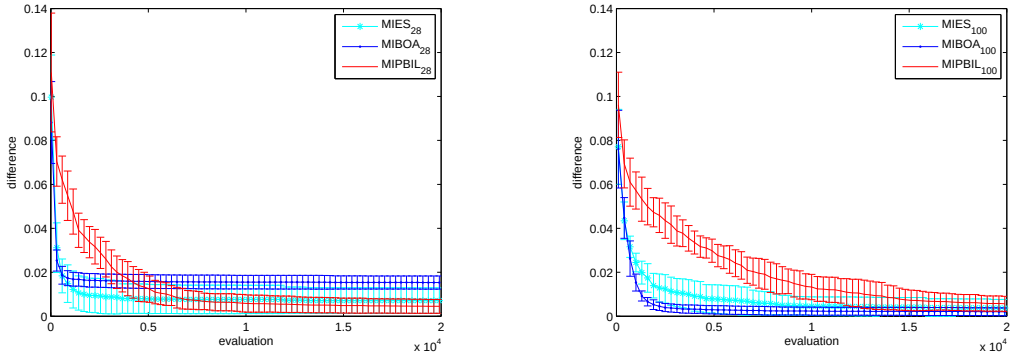


Figure 4: Convergence dynamics of MIES, MIPBIL, and MIBOA on a 'bitree'-type ADG-NKL.

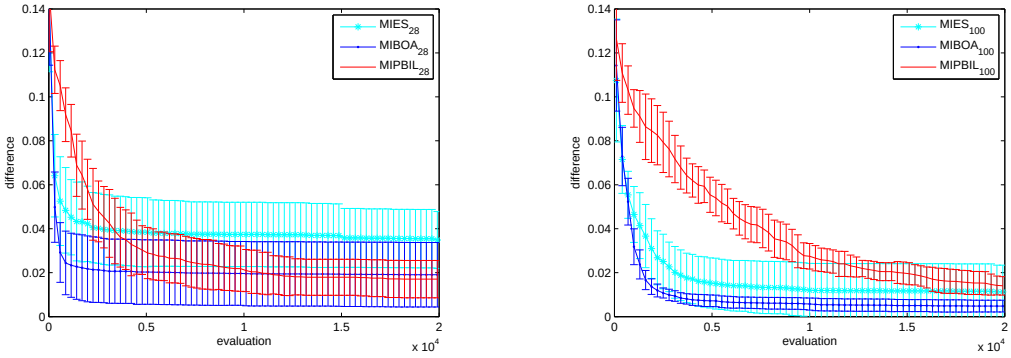


Figure 5: Convergence dynamics of MIES, MIPBIL, and MIBOA on the 'invtree'-type ADG-NKL.

Algorithm	1000 Eval.		2000 Eval.		5000 Eval.		10000 Eval.		20000 Eval.	
	Ranks	$\sum$	Ranks	$\sum$	Ranks	$\sum$	Ranks	$\sum$	Ranks	$\sum$
$MIPBIL_{28}$	555565	31	555555	30	555555	30	353545	25	354655	28
$MIPBIL_{100}$	666656	35	666666	36	666666	36	666666	36	666546	33
$MIES_{28}$	222222	12	313333	16	433334	20	434334	21	223323	15
$MIES_{100}$	444444	24	444444	24	344441	20	545452	25	545462	26
$MIBOA_{28}$	111111	6	231111	9	221223	12	222223	13	432234	18
$MIBOA_{100}$	333333	18	122222	11	112112	8	111111	6	111111	6

Table 1: Ranking position of average objective function values for $MIES_\mu$, $MIPBIL_\mu$, and $MIBOA_\mu$ with population size 28 and 100 on ADG-based Nk-landscapes after different numbers of evaluations.

iterations is reported. This table provides further evidence for the hypothesis that the introduction of the dependence information in the MIBOA is beneficial. In addition it can be observed that a small population size helps to speed up convergence of the algorithm in the short term, while a large population size improves its long term behaviour. For further details and results of this study we refer to [18].

7 Summary and Outlook

In this contribution we studied how knowledge on acyclic dependency structures can be integrated into stochastic optimization for mixed-variable search spaces. The Mixed-integer Bayesian Optimizaton Algorithm (MIBOA), an estimation of distribution algorithm working with heterogeneous Bayesian networks with a-priori set structure, was designed and studied. As a test environment mixed-integer Nk-landscapes have been modified to ADG-based mixed-integer Nk-landscapes. The dependence structure of their variables is defined as an ADG, and as a proof of concept it had to be studied, whether the MIBOA can exploit a-priori knowledge on this dependency structure or not. The test shows that the MIBOA algorithm can indeed take advantage of this a-priori information on dependences. In all cases of ADGs discussed ('chain', 'struct2', 'struct3', 'bitree', 'tritree', and 'invtree') we observed a performance gain as compared to mixed-integer evolution strategies and estimation of distribution algorithms, both working with an independent joint distribution, namely MIES and MIPBIL. The population size of MIBOA turned out to be an important parameter to control the trade-off between a fast convergence speed in the beginning and a reliable convergence to the global optimum towards the end of the search.

Future work will have to focus on studies on further synthetic and real-world problems, including cases where discrete parameters depend on continuous parameters, which turned out to be difficult to handle. In particular we are interested in applying the new algorithm in the context of optimization of image processing pipelines, the acyclic structure of which makes the MIBOA a particularly promising technique.

Acknowledgments This research is supported by the Netherlands Organization for Scientific Research (NWO).

References

[1] L. Altenberg. *The Handbook of Evolutionary Computation*, chapter NK-Fitness Landscapes. Oxford University Press, 1997.

[2] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In A. Prieditis and S. Russel, editors, *The Int. Conf. on Machine Learning 1995*, pages 38–46, San Mateo, CA, 1995. Morgan Kaufmann Publishers.

[3] B.C. Cobb, R. Rumi, and A. Salmerón. Bayesian network models with discrete and continuous variables. In *Advances in Probabilistic Graphical Models*, pages 81–102, 2007.

[4] R. G. Cowell, A. Philip Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag New York, 1999.

[5] J. Eggermont, R. Li, E.G.P. Bovenkamp, H.A. Marquering, M.T.M. Emmerich, A.v.d. Lugt, Th. Bäck, J. Dijkstra, and J.H.C. Reiber. Optimizing computed tomographic angiography image segmentation using fitness based partitioning. In *EvoIASP08*, volume 4974 of *LNCS*, pages 275–284. Springer, 2008.

[6] M. Emmerich, M. Grötzner, B. Groß, and M. Schütz. Mixed-integer evolution strategy for chemical plant optimization with simulators. In I.C.Parmee, editor, *Evolutionary Design and Manufacture - Selected papers from ACDM'00*, pages 55–67. Springer, 2000.

[7] D. Koller, U. Lerner, and D. Anguelov. A general algorithm for approximate inference and its application to hybrid bayes nets. In *Proceedings of Conference on Uncertainty in Artificial Intelligence UAI*, pages 302–313, 2007.

[8] P. Larranaga and J.A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, volume 2 of *Genetic Algorithms and Evolutionary Computation*. Kluwer Academic, 2001.

[9] S.L. Lauritzen. *Graphical models*. Clarendon Press, Oxford, 1996.

[10] R. Li, M.T.M.Emmerich, J. Eggermont, E.G.P. Bovenkamp, Th. Bäck, J. Dijkstra, and J.H.C. Reiber. Mixed-Integer NK Landscapes. In etc. T.P. Runarsson, editor, *Parallel Problem Solving from Nature - PPSN IX, 9th international Conference*, volume 4193 of *LNCS*, pages 42–51. Springer, 2006.

[11] R. Li, M.T.M.Emmerich, J. Eggermont, E.G.P. Bovenkamp, Th. Bäck, J. Dijkstra, and J.H.C. Reiber. Mixed-integer optimization of coronary vessel image analysis using evolution strategies. In *GECCO06*, pages 1645–1652, New York, NY, USA, 2006. ACM.

[12] K.P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, San Francisco, CA, 1988.

[14] M. Pelikan, D.E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 13-17 1999. Morgan Kaufmann Publishers, San Fransisco, CA.

[15] M. Schütz and J. Sprave. Application of parallel mixed-integer evolution strategies with mutation rate pooling. In *Evolutionary Programming*, pages 345–354, 1996.

[16] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.

[17] M. Sebag and A. Ducoulombier. Extending population-based incremental learning to continuous search spaces. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 418–427, London, UK, 1998. Springer-Verlag.

[18] A. Zhang. Bayesian mixed integer optimization using a-priori knowledge on variable dependences. Internal report, LIACS, Leiden University, Leiden, NL, August 2008.

From Probabilistic Horn Logic to Chain Logic

Nivea Ferreira Arjen Hommersom Peter Lucas

Institute for Computing and Information Sciences,
Radboud University Nijmegen, Nijmegen, The Netherlands
Email: {nivea,arjenh,peterl}@cs.ru.nl

Abstract

Probabilistic logics have attracted a great deal of attention during the past few years. Where logical languages have, already from the inception of the field of artificial intelligence, taken a central position in research on knowledge representation and automated reasoning, probabilistic graphical models with their associated probabilistic basis have taken up in recent years a similar position when it comes to reasoning with uncertainty. There are now several different proposals in literature to merge logic and probabilistic graphical models. Probabilistic Horn logic combines Horn logic with probability theory, which yields a probabilistic logic that allows reasoning with classes of Bayesian networks. Bayesian logic is similar in expressive power to probabilistic Horn logic; the main difference is that it is primarily meant as a language for generating Bayesian networks. Finally, Markov logic networks have recently been proposed as a language for generating Markov networks using a model-theoretic interpretation of a logical specification. However, whereas Bayesian networks have an attractive semantics, they suffer from the fact that different Bayesian networks may represent exactly the same independence relation. Markov networks, on the other hand, lack in expressiveness when representing independence information. The formalism of chain graphs is increasingly seen as a natural probabilistic graphical formalism as it generalises both Bayesian networks and Markov networks, and has an attractive semantics in the sense that any Bayesian network has a unique graphical representation as a chain graph. In this paper, a new probabilistic logic, called chain logic, is developed along the lines of probabilistic Horn logic. This new probabilistic logic allows representing subtle independence information that cannot be represented by all previously developed probabilistic logics.

1 Introduction

There has been a considerable amount of work in the field of artificial intelligence during the past two decades on integrating logic and probability theory. This research was motivated by perceived limitations of both formalisms. Logic has for long acted as the common ground for almost all research on knowledge representation and reasoning in artificial intelligence; yet, uncertainty cannot be handled easily in logic. Probabilistic graphical models have been proposed as formalisms for reasoning with uncertainty, taking probability theory as their foundation. Although their associated graphical representation allows specifying relationship among objects in the domain of discourse such that it is possible to reason about their statistical dependences and independences, probabilistic graphical models are essentially propositional in nature, and they lack the representational richness of logics.

Several researchers have proposed probabilistic logics that merge the two types of languages in an attempt to redress their individual shortcomings. A variety of such languages is now available, each of them adopting a different view on the integration. Unfortunately, it appears that all of the available frameworks are still restricted in one way or the other. For instance, probabilistic Horn logic, as originally proposed by Poole in [4], offers a framework that was shown to be as powerful as Bayesian networks, yet it has the advantage that it is a first-order language that integrates probabilistic and logical reasoning in a seamless fashion. However, usually the graphical representation associated with a Bayesian network does not offer a unique

This work has been done as part of the B-Screen project funded by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.066.605, and as part of the Octopus project under the responsibility of the Embedded Systems Institute. The Octopus project is partially supported by the Dutch Ministry of Economic Affairs under the Bsik program.

way to represent the independence information, which makes the interpretation of Bayesian networks cumbersome. Bayesian logic programs [1] have similar limitations as probabilistic Horn logic; in addition, they are only proposed as formalisms to specify Bayesian networks in a logical way and reasoning is done in the generated Bayesian networks. Finally, the framework of Markov logic networks [5] has been proposed as a powerful language based on first-order logic to specify Markov networks. Yet, Markov networks are seen by researchers in probabilistic graphical models as the weakest type of such models, as much of the subtleties of representing conditional dependence and independence cannot be handled by Markov networks.

In this paper, we propose modifications and extensions to probabilistic Horn logic, yielding a first-order language that is more expressive than the languages mentioned above, in the sense that the probabilistic models that can be specified and reasoned about have Bayesian networks and Markov networks as special cases. This new probabilistic logic is called *chain logic*.

The organisation of this paper is as follows. In Section 2 we provide an overview of the basic notions of Horn clauses and chain graphs. Section 3 contains an introduction to the chain logic language, with details on its syntax and semantics. Finally, Section 4 contains a comparison to other work and Section 5 presents our conclusions.

2 Preliminaries

2.1 Abduction Logic

Function-free Horn logic is a subset of first-order logic, whose formulae are constructed using *constants* representing individual objects in the domain, *variables* for quantifying over individuals, and *predicates* for representing relations among individuals. Predicates applied to a tuple of terms are called atomic formulae, or *atoms* for short. Formulae in Horn logic, called *Horn clauses*, have the following form:

$$D \leftarrow B_1, \ldots, B_n$$

where $D, B_1, \ldots, B_n$ are atoms, $\leftarrow$ stands for logical implication and the commas ',' stand for conjunction. Sets of Horn clauses are interpreted as conjunctions. All variables appearing in a Horn clause are *universally* quantified. D is called the *head* and $B_1, \ldots, B_n$ constitutes the *body* of the clause. When simultaneously replacing all occurrences of variables in a Horn clause ψ by constants, a so-called *ground* formula results.

Abduction logic is defined as a special variant of function-free Horn logic, where the syntax of Horn clauses is slightly modified, and $\leftarrow$ is given a causal interpretation. *Abduction clauses* have the following form:

$$D \leftarrow B_1, \ldots, B_n : R_1, \ldots, R_m$$

where the predicates of the atoms D and B_i are unary and the atoms R_j, called *templates*, express relationships among variables, where all variables appearing in the atoms D and B_i occur in at least one template R_j. Atoms that do not occur as head of a clause are called *assumables*. From a logical point of view, the ':' operator has the meaning of a conjunction; it is only included in the syntax to allow separating atoms that are templates from non-template atoms. The basic idea is to use unary predicates to represent variables (later referred to as *random* variables), and the templates R_j to represent relations among those variables.

Abduction logic has a standard model-theoretic semantic, defined in terms of the logical consequence operator $\vDash$, and a sound and complete procedural semantics, defined in terms of the deduction relation, indicated by $\vdash$. Let T be a set of abduction clauses, called an *abductive theory* in this paper, then, using model theory or deduction,concluding formula ψ from the theory is denoted by $T \vDash \psi$ and $T \vdash \psi$, respectively.

For abduction logic a special type of logical reasoning has been proposed, called *abduction*, which is defined in terms of model theory or deduction using so-called *explanations*. Let A be the set of all assumables and let A' denote the set of ground instances of A. Given a set of atoms O, interpreted as *observations*, then these observations are explained in terms of the abductive theory and a set of assumables.

Definition 1. *An* explanation *of a set of atoms O based on the pair $\langle T, A\rangle$ is defined as a set of ground assumables $H \subseteq A'$ satisfying the following conditions:*

- *$T \cup H \models O$, and*
- *$T \cup H$ is consistent, i.e., $T \cup H \nvDash \bot$.*

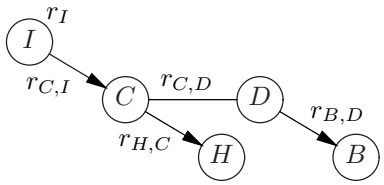


Figure 1: Causal network model of causal and associational knowledge about influenza.

A minimal explanation H *of* O *is an explanation whose proper subsets are not explanations of* O*. The set of all minimal explanations is denoted by* $\mathcal{E}_T(O)$.

Consider the following example.

Example 1. *Suppose that we have the following piece of medical knowledge. Influenza* (I) *causes coughing* (C)*, where coughing is known as a possible cause for hoarseness* (H)*. In addition, coughing is known to be associated with dyspnoea (shortness of breath)* (D)*, although a clear cause-effect relationship is missing. Dyspnoea restricts the oxygen supply to the blood circulation; the resulting low oxygen saturation of the blood will turn the skin to colour blue* (B)*, which is a condition called cyanosis. This qualitative knowledge is represented by the causal network shown in Figure 1. The associated abductive theory* T *is the following:*

$$
\begin{array}{l}
H(x) \leftarrow C(y) : r_{H,C}(x,y) \\
B(x) \leftarrow D(y) : r_{B,D}(x,y) \\
C(x) \leftarrow I(y) : r_{C,I}(x,y), r_{C,D}(x,z) \\
D(x) \leftarrow : r_{C,D}(y,x) \\
I(x) \leftarrow : r_I(x)
\end{array}
$$

where each of the variables has $\{f,t\}$ *as domain. It now holds that:*

$$T \cup \{r_I(t), r_{H,C}(t,t), r_{C,I}(t,t), r_{C,D}(t,t)\} \models H(t)$$

and $T \cup \{r_I(t), r_{H,C}(t,t), r_{C,I}(t,t), r_{C,D}(t,t)\} \not\models \bot$.

The intuition behind the syntax of abduction clauses, such as $C(x) \leftarrow I(y) : r_{C,I}(x,y), r_{C,D}(x,z)$*, is that* $C(x) \leftarrow I(y)$ *expresses the* potential *existence of a causal relation between the referred atoms, here* $I(y)$ *and* $C(x)$*. Templates* R_i*, e. g.* $r_{C,I}(x,y)$*, expresses whether the relationship actually does or does not hold. When there are no atoms to the left of the ':' operator, such as in the clause* $D(x) \leftarrow : r_{C,D}(z,x)$*, the template represents an association rather than a causal relation.*

2.2 Chain Graphs

A chain graph (CG) is a probabilistic graphical model that consists of labelled vertices, that stand for random variables, connected by directed and undirected edges. This graphical representation allows chain graphs to be considered as a framework that generalises both acyclic directed graph (ADG) models, i.e., Bayesian networks, and undirected graph (UG) models, i.e., Markov networks [3]. The definitions with respect to chain graphs given in this paper are in accordance with Lauritzen [2].

Let $G = (V, E)$ be a *hybrid graph*, where V denotes the set of *vertices* and E the set of *edges*, where an edge is either an *arc* (directed edge), or a *line* (undirected edge). Let indexed lower case letters, e.g., v_1 and v_2, indicate vertices of a chain graph. We denote an arc connecting two vertices by $\rightarrow$ and a line by '$-$'. Consider two vertices v_1 and v_2. If $v_1 \rightarrow v_2$ then v_1 is a *parent* of v_2. If $v_1 - v_2$ then v_1 is a *neighbour* of v_2. The set of parents and neighbours of a vertex v are denoted by $\mathrm{pa}(v)$ and $\mathrm{ne}(v)$, respectively.

A *path* of length n in a hybrid graph $G = (V, E)$ is a sequence of distinct vertices $v_1, \ldots, v_n$, such that either $v_i - v_{i+1} \in E$ or $v_i \rightarrow v_{i+1} \in E$. A *directed path* is a path which includes at least one arc, and where all arcs have the same direction. A *cycle* is a path where the first and last vertex are the same. A *chain graph* is a hybrid graph with the restriction that no directed cycles exist.

If there is an edge between every pair of vertices in a set of vertices, than this set is named *complete*. A *clique* is a maximally complete subset. Now, consider the graph obtained from a chain graph by removing all its arcs. What are left are vertices connected by lines, called *chain components*; the set of all chain components is denoted here by $\mathcal{C}$.

Associated to a chain graph $G = (V, E)$ is a joint probability distribution $P(X_V)$ that is faithful to the chain graph G, i.e., it includes all the independence information represented in the graph. This is formally expressed by the following *chain graph Markov property*:

$$P(X_V) = \prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) \tag{1}$$

with $V = \bigcup_{C \in \mathcal{C}} C$, and where each $P(X_C \mid X_{\mathrm{pa}(C)})$ factorises according to

$$P(X_C \mid X_{\mathrm{pa}(C)}) = Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \tag{2}$$

given that $M(C)$ is the complete set in the moral graph[1] obtained from the subgraph $G_{C \cup \mathrm{pa}(C)}$ of G. The functions φ are real positive functions, called *potentials*; they generalise joint probability distributions in the sense that they need not be normalised.

Finally, the normalising factor Z is defined as

$$Z(X_{\mathrm{pa}(C)}) = \sum_{X_C} \prod_{M \in M(C)} \varphi_M(X_M) \tag{3}$$

As a Bayesian network is a special case of a chain graph model, Equation (1) simplifies in that case to:

$$P(X_V) = \prod_{v \in V} P(X_v | X_{\mathrm{pa}(v)}) \tag{4}$$

which is the well-known factorisation theorem of Bayesian networks [2]. In this case, the chain components are formed by a family of random variables. Therefore, for each of those random variables the distribution is defined as the conditional probability function of this variable, given the value of its parents. Note that according to Equation (1), chain graphs can also be interpreted as an ADG of chain components.

3 Chain Logic

3.1 Language Syntax

The formalism presented in this section is inspired by probabilistic Horn logic as introduced by Poole in [4]. For the sake of simplicity, we assume here finite domain specifications. Furthermore, the unique names assumption holds for the different constants of the domain.

Chain logic (CL) extends abduction logic as described in Section 2.1 by interpreting templates as representing uncertain events. The actual definition of the uncertainty is done by means of a *weight* declaration. This is of the form

$$weight(a_1 : w_1, \ldots, a_n : w_n) \tag{5}$$

where a_i represents an atom and $w_i \in \mathbb{R}_0^+$. The set of atoms appearing in such declarations are the assumables, denoted by A. Here we assume that the atoms in a weight declaration share the same variables. For a grounded assumable a, the use of function $\omega(a)$ defines the weight w that is associated to this assumable. We require that a ground atom a – which is an instance of one of the assumables – does not appear as an instance of another assumable in another weight declaration.

In short, the weight declaration defines conjunctions of atoms that are mutually exclusive and exhaustive. Therefore, together with the above elements, a CL specification also includes integrity constraint statements. For instance, clauses of the form

$$\bot \leftarrow a_i \wedge a_j \tag{6}$$

for any pair a_i and a_j appearing in the same weight declaration where $i \neq j$. Such clauses are implicit in all of our given examples. We also allow the addition of another set of constraints referring to a pair of assumables appearing in different weight declarations, as seen in the example below.

[1] Moralisation encompasses: (1) adding lines between unconnected parents of a chain component, and (2) conversion of arcs into lines by ignoring their directions.

Example 2. *Consider the description given in Example 1. Uncertainty is defined by replacing the templates by potential functions. For the abductive theory in this example, they are as follows:*

φ_{CI}	i	$\bar{i}$
c	4	2
$\bar{c}$	1	10

φ_{CD}	d	$\bar{d}$
c	18	2
$\bar{c}$	5	2

φ_{HC}	c	$\bar{c}$
h	0.6	0.1
$\bar{h}$	0.4	0.9

φ_{BD}	d	$\bar{d}$
b	0.3	0.001
$\bar{b}$	0.7	0.999

φ_I	
i	0.1
$\bar{i}$	0.9

This example can be represented in chain logic using the following abduction clauses:

$$\begin{array}{ll} I(x) \leftarrow\!: \varphi_I(x) & C(x) \leftarrow I(y) : \varphi_{CI}(x,y) \wedge \varphi_{CD}(x,z) \\ D(y) \leftarrow\!: \varphi_{CD}(x,y) & H(x) \leftarrow C(y) : \varphi_{HC}(x,y) \\ B(x) \leftarrow D(y) : \varphi_{BD}(x,y) & \bot \leftarrow \varphi_{CI}(x,y) \wedge \varphi_{CD}(\bar{x},z) \end{array}$$

Furthermore, we can associate weights to the assumables according to the potential functions. For instance, considering φ_{CD}, we have:

$$weight(\varphi_{CD}(t,t) : 18, \varphi_{CD}(t,f) : 2, \varphi_{CD}(f,t) : 5, \varphi_{CD}(f,f) : 2)$$

In order to be able to probabilistically interpret a CL theory T, a number of assumptions are added to those of abduction logic:

1. the theory is acyclic: if T' is the set of ground instances of elements of T, it is possible to assign a natural number to every ground atom such that for every rule in T' the atoms in the body of the rule are strictly less than the atom in the head;
2. the rules for every ground non-assumable represented in T' are covering, i.e., there is always a rule whose assumable holds – which is used as an explanation of the atom in the head;
3. the bodies of the rules in T' for an atom are mutually exclusive;
4. there is a set of ground assumables, one from every grounded weight declaration, consistent with T.

As in Poole's probabilistic Horn logic, these assumptions are not intended to be enforced by the system: it is up to the modeller to comply to these requisites. Under this condition, we can then guarantee the probabilistic properties of the theory, as we show in the next section.

3.2 Probabilistic Reasoning

In this section, we show how we can infer probabilities from a chain logic theory. A conjunction of ground instances, one for each assumable of a grounded weight declaration, is called a *state*. The set of all such states is denoted by S. The set of *consistent states*, with respect to a theory T, will be denoted by CS, i.e., $\mathrm{CS} = \{s \in S \mid T \cup \{s\} \nvDash \bot\}$. The last assumption mentioned in the previous section can be expressed formally by $\mathrm{CS} \neq \varnothing$.

Definition 2. *Let P_T be a real positive function of S that is defined as follow:*

$$P_T(s) = \begin{cases} \frac{1}{Z} \prod_{a \in s} \omega(a) & \text{if } s \in \mathrm{CS} \\ 0 & \text{otherwise} \end{cases}$$

where $Z = \sum_{s \in \mathrm{CS}} \prod_{a \in s} \omega(a)$.

Clearly, the function P_T obeys the axioms of probability theory, as each weight is larger than or equal to 0 and, given that $\mathrm{CS} \neq \varnothing$, it follows that $\sum_{s \in S} P_T(s) = 1$, and is thus a joint probability distribution; P_T is sometimes abbreviated to P in the following.

Given T, a minimal explanation e of some formula ψ is equivalent to a disjunction of consistent states, i.e., $T \models e \equiv \bigvee_i s_i$ with $s_i \in \mathrm{CS}$. As all s_i are mutually exclusive, it follows that:

$$P_T(e) = P_T(\bigvee_i s_i) = \sum_i P_T(s_i)$$

which offers the means to assign a probability to minimal explanations. In order to assign a probability to a formula, we have the following result.

Theorem 1. *Under the assumptions mentioned in Section 3.1, if $\mathcal{E}_T(\psi)$ is the set of minimal explanations of the conjunction of atoms ψ from the chain logic theory T, then:*

$$P_T(\psi) = \sum_{e \in \mathcal{E}_T(\psi)} P(e)$$

Proof. This follows exactly the same line of reasoning of [4, page 53, proof of Theorem A.13]. □

This result shows that P is indeed a probability distribution over conjunctions of formulae if we use the definition of P_T above. Other probabilities can be calculated on the basis of these types of formulae, such as conditional probabilities which can be calculated according to the definition of conditional probabilities $P(a|b) = \frac{P(a \wedge b)}{P(b)}$. Below, we will sometimes refer to the resulting probability distribution by P_T in order to stress that we mean the probability calculated using Definition 2.

Example 3. *Reconsider the uncertainty specification concerning influenza as described in Example 2. We will illustrate how probabilities can be calculated from the explanations obtained from an abductive scheme. Consider here that we are interested in calculating the $P(B(t))$ (i.e., the probability of B assumes truth-value t). Recalling the definitions provided in Section 2.1, we obtain the minimal explanations for $B(t)$, i.e., $\mathcal{E}_T(B(t))$ as the set with the following 4 members:*

$$\begin{array}{ll} \{\varphi_{BD}(t,t), \varphi_{CD}(t,t)\} & \{\varphi_{BD}(t,t), \varphi_{CD}(f,t)\} \\ \{\varphi_{BD}(t,f), \varphi_{CD}(t,f)\} & \{\varphi_{BD}(t,f), \varphi_{CD}(f,f)\} \end{array}$$

We can then sum over the states that are consistent with these explanations (by extending the explanations with consistent instances of φ_I, φ_{CI}, and φ_{HC}[2]):

$$P(B(t)) = \sum_{e \in \mathcal{E}_T(D(t))} = Z^{-1}(0.3 \cdot 18 \cdot 0.1 \cdot 4 \cdot 1 + 0.3 \cdot 18 \cdot 0.9 \cdot 2 \cdot 1 + 0.3 \cdot 5 \cdot 0.1 \cdot 1 \cdot 1 \ldots) = 25.756/Z$$

Notice that explanations are extended in order to incorporate other consistent influences. For instance, reasoning about the probability of B being true might only include φ_{BD} in the explanations. However, φ_{CI} and φ_I – which are relevant for such computation – are also taken into account.

3.3 Specification of Chain Graphs

In this section, we present the formal relation between chain graphs with discrete random variables and chain logic. For the sake of simplicity, we focus on chain graphs with binary variables, i.e., the set of constants is $\{t, f\}$, although the theory generalises to arbitrary arities. Complementary constants are denoted with a bar, i.e., $\bar{t} = f$ and $\bar{f} = t$.

The translation from a chain graph G to a chain logic theory T is as follows. Consider a vertex v in G. For each component $C \in \mathcal{C}$ of G, there is a set of potential functions defined on the moral graph of the sub-graph $G_{C \cup \mathrm{pa}(C)}$ which contains v. This set of potential functions is denoted by $\Phi(G, C, v)$. For every vertex v, we have the following formula in T:

$$V(x) \leftarrow \bigwedge \{V'(x_{v'}) \mid v' \in \mathrm{pa}(v)\} : \bigwedge \{\varphi_M(x_1, \ldots, x, \ldots, x_n) \mid \varphi_M \in \Phi(G, C, v)\}$$

and we ensure that each of the predicates defined for the same random variable shares that variable in the formula. However, this is not strictly necessary as different values for the same random variable in a component is also disallowed by the integrity constraints.

The integrity constraints are defined as follows. If we have two potential functions, namely an n-ary $\varphi_M(\ldots, v, \ldots)$ and an m-ary $\varphi'_M(\ldots, v, \ldots)$, i.e., which share a variable v *in the same chain component* (i.e., not between chain components), then we add the following formula to T:

$$\bot \leftarrow \varphi_M(x_0, \ldots, x, \ldots, x_n) \wedge \varphi'_M(x'_0 \ldots, \bar{x}, \ldots, x'_m)$$

for each variable that they share. As mentioned earlier, this ensures we do not generate explanations which have inconsistent assignments to the random variables within the same chain component.

Finally, for each potential function φ_M, we define a weight declaration containing $\varphi_M(c_0, \ldots, c_n) : w$ if $\varphi_M(X_M = (c_0, \ldots, c_n)) = w$.

[2] Notice that $\{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t) \cdot \varphi_{HC}(t,t)\} + \{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t) \cdot \varphi_{HC}(f,t)\} = \{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t)\}$, as $\varphi_{HC}(t,t) + \varphi_{HC}(f,t) = 1$.

Example 4. *Consider again our influenza domain example. Consider further that we are interested in the probability of $P(I(t) \mid B(t))$ (i.e., the conditional probability of I being true given that B is true). This probability can be obtained by the having $P(I(t) \wedge B(t))$ divided by $P(B(t))$. The calculation of $P(B(t))$ was shown in Example 3. Therefore, we follow by calculation the minimal explanations for $I(t) \wedge B(t)$, i.e., $\mathcal{E}_T(I(t) \wedge B(t))$ is a set with the following 4 members:*

$$\{\varphi_{BD}(t,t), \varphi_{CD}(t,t), \varphi_I(t)\} \qquad \{\varphi_{BD}(t,t), \varphi_{CD}(f,t), \varphi_I(t)\}$$
$$\{\varphi_{BD}(t,f), \varphi_{CD}(t,f), \varphi_I(t)\} \qquad \{\varphi_{BD}(t,f), \varphi_{CD}(f,f), \varphi_I(t)\}$$

Following the same reasoning as before, we obtain that $P(I(t) \wedge B(t)) = 2.32/Z$. Finally, we have that $P(I(t) \wedge B(t))/P(B(t) = (2.32/Z)/(25.756/Z) \simeq 0.0901$[3]*.*

In the following theorem, we establish that probabilities calculated from the chain logic theory corresponds to the chain graph semantics.

Theorem 2. *Suppose $v_1, \ldots, v_n$ are vertices in a chain graph, with T as the corresponding chain logic theory by the translation described above, then:*

$$P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) = P_T(V_1(c_1), \ldots, V_n(c_n))$$

Proof. There is only one minimal explanation of $V_1(c_1) \wedge \cdots \wedge V_n(c_n)$, namely $\varphi_M(c_0^M, \ldots, c_m^M)$ for all potential functions in cliques in the moral graphs of chain components with their parents, such that the constants filled into the potential functions correspond to the values for each of the random variables.

The explanation describes exactly one state. Denote this as s. As the potential functions are related to exactly one component, we have the following equation:

$$\prod_{a \in s} \omega(a) = \prod_{C \in \mathcal{C}} \prod_{\varphi_j^C(c_0^j, \ldots, c_n^j) \in s} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) = \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) \tag{7}$$

where φ^C are potential functions defined for component C and $M(C)$ are the complete sets in the moral graph from the sub-graph $G_{C \cup \mathrm{pa}(C)}$.

Let $Z = \sum_{s \in \mathrm{CS}} \prod_{a \in s} \omega(a)$. Since there are no integrity constraints between variables in chain components (i.e., combinations of consistent potential functions which are in different chain components are consistent), we have that:

$$Z = \sum_{s \in \mathrm{CS}} \prod_{a \in s} \omega(a) = \prod_{C \in \mathcal{C}} \sum_{s \in \mathrm{CS}(C)} \prod_{\varphi_j^C(c_0^j, \ldots, c_n^j) \in s} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) = \prod_{C \in \mathcal{C}} Z(X_{\mathrm{pa}(C)}) \tag{8}$$

where $\mathrm{CS}(C)$ is the set of consistent states (w.r.t. T) restricted to the potential functions in that chain component. Then, the equivalence follows in the following way:

$$\begin{aligned}
P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) &= \textstyle\prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) && \text{(factorisation)} \\
&= \textstyle\prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \cdot && \text{(factorisation)} \\
&= \left(\textstyle\prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)})\right) \textstyle\prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) && \text{(arithmetic)} \\
&= Z^{-1} \textstyle\prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M)) && \text{(Eq. 8)} \\
&= Z^{-1} \textstyle\prod_{a \in w} \omega(a) && \text{(Eq. 7)} \\
&= P_T(V_1(c_1), \ldots, V_n(c_n))) && \text{(def. } P_T\text{)}
\end{aligned}$$

□

As we have shown in Section 3.2 that P_T adheres to the axioms of probability theory, chain graphs and the translated chain logic theory agree on all probabilities. This result shows that chain graphs can be translated to chain logic specifications. The converse is also true: all chain logic programs, which adhere to the assumptions of Section 3.1, correspond to a chain graph – at least in a trivial sense – as a fully connected Markov network models and the associated probability distributions, which can be derived from the chain logic semantics. However, the independence information that is implicit in the chain logic specification will not be represented.

[3] Even though not used here, remember that Z is calculated according to the set of consistent states, i. e., all the possible instantiations of $\{t, f\}$ in the potential functions which satisfy the integrity constraints and follows domain descriptions.

4 Related Work

As mentioned in Section 3.1, the language presented here is inspired by Poole's probabilistic Horn logic [4]. Besides some changes in the terminology (such as using *weight* declarations in place of *disjoint* ones), the main differences in terms of syntax is the set of integrity constraints allowed and the probabilistic information captured in each formalism. Also, in probabilistic Horn logic the disjoint declarations should sum up to 1, whereas weights can sum up to any value. This enabled the formalisation of potential functions instead of a (normalised) probability distribution. In terms of independence, in Poole's definition instantiations of hypotheses that appear in different disjoint declarations are independent. In our case, by allowing the use of extra integrity constraints, we are able to establish dependences among such hypotheses (cf. Example 2).

In fact, those differences extend Poole's approach and allow us to obtain a more generic probabilistic model, being crucial for the representation of chain graph models. By using potential functions we can represent the quantitative influence between variables in a clique. The additional integrity constraints guarantee that instantiations of those potentials functions appear consistently in each explanation.

Despite such differences, we still share with Poole's approaches some assumptions and similar results, for instance, with respect to the probability densities defined over the theory. One additional assumption in chain logic, namely assumption (4) in Section 3.1, is not present in probabilistic Horn logic since this property is true for any probabilistic Horn logic theory.

As a first-order language-based formalism for probabilistic graphical models, we can also relate our work to, for instance, Bayesian logic programs [1] and Markov logic [5]. We present here a simple language that can be used for the specification of both Bayesian and Markov network models, in such a way that the logical specification is more that a generative language for the model at hand, maintaining a close relation between logical and probabilistic reasoning – without loss of expressiveness.

5 Final Considerations

In this paper we presented a simple and yet powerful language for specifying and reasoning about chain graphs. Besides being able to incorporate both Bayesian and Markov network models as special cases, we maintain a strong relation between logical and probabilistic reasoning.

Our language still presents some restrictions. First, we use finite set of constants, which prohibits the use of continuous variables. For Markov logic networks, it has been shown that special cases of such networks can be extended to infinite domains by defining a Gibbs measure over sets of interpretations for logical formulae [6]. A similar approach could be taken here by defining a measure over the set of consistent states. Another limitation is the acyclicity assumption, which restricts the explicit representation of undirected graphs components. Even though we require certain assumptions for a sound probabilistic interpretation, weakening acyclicity seems feasible.

While we have shown in this paper that chain logic is powerful enough to define and reason about chain graphs, we have no strong reason to suspect that chain logic is restricted to this class of probabilistic graphical models. While chain graphs is a fairly general class of graphs, it might be the case that the language is applicable to a broader set of graphs. Furthermore, modelling the independence implied in chain logic theories into a graphical model is an open question that will be investigated further.

References

[1] K. Kersting and L. de Raedt. Bayesian logic programs. Technical Report 151, Institute for Computer Science - University of Freiburg, 2001. CoRR cs.AI/0111058.

[2] S. L. Lauritzen. *Graphical Models.* Oxford:Clarendon, 1996.

[3] J. Pearl. *Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] D. Poole. Probabilistic Horn abduction and Bayesian networks. *AI Journal*, 64(1):81–129, 1993.

[5] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[6] P. Singla and P. Domingos. Markov logic in infinite domains. In *Proc. of UAI'07*, pages 368–375. AUAU Press, 2007.

Visualizing Co-occurrence of Self-Optimizing Fragment Groups

Edgar H. de Graaf Walter A. Kosters

Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
email: edegraaf@liacs.nl

Abstract

In this paper we will use and extend a type of competitive neural network to visualize the co-occurrence of subgraphs in a dataset where molecules are considered as transactions or records.

We will adapt this algorithm, called push-and-pull, to plot each subgraph as a point in 2D space by repeatedly changing its position based only on the relative distances. Instead of each fragment/subgraph being a point ($\{x, y\}$-coordinate) in our visualization, we build a group for each point by "leaking" fragments to a point linked to a more fitting group.

In this way we create a 2D visualization by knowing only the distances between a user-defined number of groups (points), improving runtime and overview. Practically it will allow us to improve the analysis of co-occurring substructures, or fragments, within the molecules of the transactions, by improving the visualization with less points (centroids) in comparison with traditional push-and-pull.

The algorithm is beneficial for any data mining task where one only knows the distances between points, because the structure of the learning examples does not clearly allow for an input vector (e.g., graphs or trees), the dimension of the input vector grows exponentially or the input vectors are simply not given.

1 Introduction

Unsupervised learning methods allow us to visualize all kinds of bio-chemical data. The motivation for this paper is the search for co-occurring substructures in sets of molecules. These substructures are basically connected subgraphs of the bigger connected graph, the *molecule*. In the context of bio-chemical data we call these subgraphs *fragments*. A visualization of co-occurrence needs to handle many fragments, we seek to improve scalability.

For a bio-chemist it is very interesting to know which fragments occur often together, for example in so-called active molecules. This is because frequent co-occurrence implies that the fragments are needed simultaneously for biological activity. Furthermore, pharmaceutical companies provide generated libraries of molecules. A visualization of co-occurrences in molecule libraries gives a bio-chemist insight how the libraries are constructed by the company.

In this paper we will use and extend a type of competitive neural network called the *push-and-pull* algorithm, as published in [9], to visualize the co-occurrence of subgraphs in a dataset where molecules are considered as transactions. The push-and-pull algorithm is related to multi-dimensional scaling for which a rich literature exists, e.g., Sammon et al. in [16], Bronstein et al. in [1], and the ISOMAP algorithm in [17].

We will adapt the push-and-pull algorithm, that plots each point in 2D space by changing its position based only on the pair-wise relative distance. A *point* in this paper is defined as $\{x, y\}$-coordinate linked to or belonging to a fragment. Instead of each fragment being a point in our visualization, now we build a group for each point. We call a point linked with a group a *centroid* since the distance between groups is decided by the average distance between members. Furthermore we make these groups *self-optimizing* by "leaking" fragments to a centroid with a more fitting group.

The goal of this research was to make the push-and-pull algorithm more scalable to the number of fragments. An added benefit is that having less points improves the overview in the visualization. With

our algorithm we hope to combine the advantages of Self-Organizing Maps (see [8]), e.g., one neuron for many similar points, with the advantages of push-and-pull, e.g., good intermediate approximations and visualization with only distance information. To this end, this paper makes the following contributions:
— We will **define the "leaking" bi-dimensional centroids** and show how they fit in the push-and-pull algorithm.
— Furthermore we will **propose an algorithm** that allows us to visualize the co-occurrence of fragments (or subgraphs).
— We will empirically show that the algorithm can rediscover **2D synthetical data** based only on their relative distance.
— Finally we will empirically show **how scalability improves** in comparison with traditional push-and-pull.

In theory we can create a 2D visualization by knowing only the distances between a user-defined number of groups (points), improving runtime and overview. Practically it will allow us to improve the analysis of co-occurring fragments within the molecules of the transactions, by improving the visualization with less points (centroids).

The algorithm is beneficial for any data mining task where one only knows the distances between points, because the structure of the learning examples does not clearly allow for an input vector (e.g., graphs or trees), the dimension of the input vector grows exponentially or the input vectors are simply not given. In these cases more traditional algorithms, e.g., K-means in [11] and EM in [2], are harder to use (since we know only the distance between instances and not their corresponding input vector).

Fast access of the distances is important for the push-and-pull algorithm as it is for any competitive neural network algorithm. The algorithm will benefit from faster on-demand distance computation, since storing all distances in memory or even on the disk becomes impractical as the number of data points grows.

The overview of the rest of the paper is as follows. In Section 2 we start with a background discussion and continue, in Section 3, with defining molecules and fragments as graphs and subgraphs and the distance measure of co-occurrence for the fragments. In Section 4 we continue with the main contribution of this paper when we define our model and the principle of leaking, introducing our extended algorithm for visualization in Section 5. Finally, in Section 6, we discuss our experimental results. We conclude in Section 7.

2 Background

This work is related to work done on competitive neural networks, more specifically involving the push-and-pull algorithm. Furthermore, the work is related to work done on the analysis of molecular datasets.

Competitive neural networks in the area of biology are important because the dimensional reduction property provides an important basis for any visualization. In general our work is related to SOMs as developed by Kohonen (see [8]), in the sense that SOMs are also used to visualize data through a distance measure. A *Self-Organizing Map* (SOM) is a type of artificial neural network that is trained to produce a low dimensional representation of the training samples. A SOM is constructed by moving the best matching point and its neighbours (within a lattice of neurons) towards the input node.

SOMs have been used in a biological context many times, for example in [6, 12]. In some cases molecules are clustered via numeric data describing each molecule; in [19] clustering such data is investigated.

Points in a SOM can not, beforehand, be linked with a single pattern or group of patterns, which makes it less suitable for our purposes. Also by design a SOM needs to know the dimension of the input vector. In our setting we do not know the input vector, but only the distance between points. One could use the distance to other points as an input vector, but the number of dimensions will potentially be huge. With a more scalable push-and-pull algorithm we will not have these disadvantages while still being able to analyse many fragments. Furthermore the algorithm for a SOM needs time before neurons get into the neighbourhood of the correct group of items. The intermediate picture of push-and-pull always approaches the situation with different levels of quality.

Furthermore, our work is related to work done on the identification of *Structure Activity Relationships* (SARs) where one relates biological activity of molecules by analyzing their chemical structure [4, 7] in the sense that in our work the structure of a graph is used to build a model. In [3, 14, 15] a statistical analysis was done on the presence of fragments in active and inactive molecules. However, our work is not concerned with the discovery of SARs, but with co-occurrence of subgraphs occurring in a collection of graphs. More related is the work done by Lameijer et al. in [10]. This work is concerned with co-occurring fragments discovered with a graph splitting. Graph splitting breaks molecules at topologically interesting points. A

frequency threshold is used to filter out some fragments after their generation, however no frequent pattern mining techniques are used. Furthermore, they do not build a co-occurrence model or a similar visualization of co-occurrence. Figure 1 shows two co-occurring subgraphs (fragments) discovered by Lameijer et al. in their dataset of molecules. The algorithm presented in this paper confirms these results.

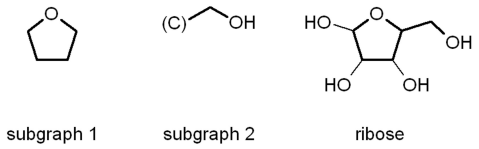


Figure 1: An example of co-occurring subgraphs from [10] with an example molecule.

3 Molecules and Fragments

First we define what a fragment and a molecule are in the context of this work. Let $G = (V, E)$ and $G' = (V', E')$ be connected graphs, where V and V' are finite, non-empty sets of vertices and E and E' are non-empty sets of edges (links between pairs of vertices). The graph G' is a *subgraph* or *fragment* of the graph or *molecule* G if $V' \subseteq V$ and $E' \subseteq E$. If G' is a subgraph of at least $minsupp$ graphs G in a dataset $\mathcal{D}$ of graphs then we call G' a *frequent subgraph*, where $minsupp$ is a user-defined threshold for frequency. Our algorithm is commonly used in the analysis of frequent fragments in order to minimize the pattern space.

The distance function for calculating the distance between two learning examples can be different for each problem, however it needs to range between 0 and 1. In the case of mining molecular datasets and analysing co-occurrence, we take the distance measure given in [5]:

$$g\_dist(g_1, g_2) = \frac{supp(g_1) + supp(g_2) - 2 \cdot supp(g_1 \wedge g_2)}{supp(g_1 \vee g_2)} \tag{1}$$

where g_1 and g_2 are two subgraphs (or fragments) and the *support* function value $supp(g)$ computes the number of occurrences of g as subgraph in the dataset of molecules. Here $supp(g_1 \vee g_2)$ counts the occurrences of one of the two graphs and $supp(g_1 \wedge g_2)$ the occurrences of both graphs. For each molecule from the dataset we count only one occurrence. If $supp(g_1 \vee g_2) = 0$ we define $g\_dist(g_1, g_2) = 1$.

This distance measure is known as the Jaccard metric and was primarily chosen for its common use in Bio-informatics (see [18]). It is also easy to compute, given the appropriate supports; it doesn't make use of complicated graph comparisons, that would slow down the process.

4 Leaking Centroids

We will visualize co-occurrence by positioning a user-defined number n of centroids in a 2-dimensional area, where the ordered set of all centroids is indicated with $\mathcal{C}$. In this work a centroid is defined as a point linked to a group of fragments instead of to one fragment. Each centroid has coordinates within this 2-dimensional area consisting of a pair of two real numbers $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

At first, as done in traditional push-and-pull, centroids are placed at random in the 2-dimensional area. At each iteration a random pair of centroids i and j ($0 \leq i, j < n$) is selected and their relative position is changed with the following formulas:

$$\begin{aligned}
x_{\mathcal{C}_i} &\leftarrow x_{\mathcal{C}_i} - \alpha \cdot (eucl_dist(\mathcal{C}_i, \mathcal{C}_j) - m_dist(\mathcal{C}_i, \mathcal{C}_j)) \cdot (x_{\mathcal{C}_i} - x_{\mathcal{C}_j}) \\
y_{\mathcal{C}_i} &\leftarrow y_{\mathcal{C}_i} - \alpha \cdot (eucl_dist(\mathcal{C}_i, \mathcal{C}_j) - m_dist(\mathcal{C}_i, \mathcal{C}_j)) \cdot (y_{\mathcal{C}_i} - y_{\mathcal{C}_j}) \\
x_{\mathcal{C}_j} &\leftarrow x_{\mathcal{C}_j} + \alpha \cdot (eucl_dist(\mathcal{C}_i, \mathcal{C}_j) - m_dist(\mathcal{C}_i, \mathcal{C}_j)) \cdot (x_{\mathcal{C}_i} - x_{\mathcal{C}_j}) \\
y_{\mathcal{C}_j} &\leftarrow y_{\mathcal{C}_j} + \alpha \cdot (eucl_dist(\mathcal{C}_i, \mathcal{C}_j) - m_dist(\mathcal{C}_i, \mathcal{C}_j)) \cdot (y_{\mathcal{C}_i} - y_{\mathcal{C}_j})
\end{aligned} \tag{2}$$

Here α ($0 \leq \alpha \leq 1$) is the user-defined learning rate. It should not be chosen too big for not making too large adaptation towards the pair-wise distance and never converging to a good approximation of distances between all points. An α of around 0.1 was found to be a good choice in most cases. The function $eucl\_dist$ calculates the Euclidean distance between centroid coordinates and $m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_j})$ is as defined below.

This is a kind of push-and-pull algorithm which yields a visualization in which the distances in 2D correspond to distances in the pattern space. This approximation emerges due to the small adaptation of pair-wise distances. In a post processing step values are scaled to fit in coordinates ranging from 0 to 1. Note that in [9] the push-and-pull was shown to converge to an approximated 2D model of the relative distance between points.

Note that we always have a visualization: the longer we run the algorithm, the better the Euclidean distances correspond to the distances between centroids in the model. As is common to this type of algorithm, one might converge to a local minimum. However, in practice this seems to occur hardly ever.

Now we further refine our notion of centroids:

1. Each centroid $\mathcal{C}_i$ has one coordinate pair $(x_{\mathcal{C}_i}, y_{\mathcal{C}_i})$.
2. Each of them has a unique set $I_{\mathcal{C}_i}$ of learning examples (subgraphs in the molecular setting). The $I_{\mathcal{C}_i}$'s are mutually disjoint.

The distance between centroids $\mathcal{C}_i$ and $\mathcal{C}_j$ is decided by the *average distance* between the items of $I_{\mathcal{C}_i}$ and $I_{\mathcal{C}_j}$:

$$m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_j}) = \frac{\sum_{a \in I_{\mathcal{C}_i}} \sum_{b \in I_{\mathcal{C}_j}} g\_dist(a, b)}{|I_{\mathcal{C}_i}| \cdot |I_{\mathcal{C}_j}|}$$

Practically we store these distances in a $|\mathcal{C}| \times |\mathcal{C}|$ distance matrix where only $|\mathcal{C}|(|\mathcal{C}| - 1)/2$ distances are stored, since $m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_j}) = m\_dist(I_{\mathcal{C}_j}, I_{\mathcal{C}_i})$. In this way we save memory, in comparison with push-and-pull where we have one point for each fragment. Each centroid has two-dimensional coordinates and n-dimensional distance vector and that is why we call the *centroids bi-dimensional*.

We say that a centroid has a *"leaking" opportunity* if the learning examples $I_{\mathcal{C}_i}$ have a chance to be transferred to another centroid because it better fits with the items of this centroid. This requires an adaptation of the distance matrix without recalculating the distances between other items of each centroid; this is more formally described in Section 5. The term "leaking" and "updating", as seen in traditional clustering, have some similarities. However "updating" is usually done using known input vectors to update the position of the cluster. With "leaking" we move an instance to a better group and in the process the distance of this group to all other groups changes.

5 The Algorithm

The basis of our algorithm is the random placement of the centroids and interchanging iterations of model (distance) optimization and leaking of the centroids. For both iterations we can set the number of iterations, however these should not be set too high. This is because centroids should get the opportunity to exchange items and via pushing and pulling be able to adapt the model to the new situation. Formally we define this basis as Algorithm 1 (next page).

The affected distances are all distances between the centroids $\mathcal{C}_1$ and $\mathcal{C}_2$ and all other centroids (including the distance between $\mathcal{C}_1$ and $\mathcal{C}_2$). In Figure 2 we give an example distance matrix for a model of 9 centroids, where we only need to store the white area. E.g., if an item leaks from centroid 3 to 5 then the arrows indicate which distance values need to be adapted.

We do not need to completely recalculate the affected distances, instead we can update the "old" distance. Indeed, if the centroid $\mathcal{C}_i$ *loses* an item a to a centroid $\mathcal{C}_j$ then for each distance with another centroid $\mathcal{C}_k$, where $k \neq i$ and $k \neq j$:

$$m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_k}) \leftarrow \frac{m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_k}) \cdot |I_{\mathcal{C}_i}| - m\_dist(\{a\}, I_{\mathcal{C}_k})}{|I_{\mathcal{C}_i}| - 1} \tag{3}$$

$$m\_dist(I_{\mathcal{C}_j}, I_{\mathcal{C}_k}) \leftarrow \frac{m\_dist(I_{\mathcal{C}_j}, I_{\mathcal{C}_k}) \cdot |I_{\mathcal{C}_j}| + m\_dist(\{a\}, I_{\mathcal{C}_k})}{|I_{\mathcal{C}_j}| + 1} \tag{4}$$

Algorithm 1 Leaking Centroid Algorithm: CENTROIDLEAK

Require: set $\mathcal{C}$ of centroids, database $\mathcal{D}$ of items
1: Divide all $i \in \mathcal{D}$ evenly among the centroids
2: **for all** $0 \leq j < |\mathcal{C}|$ **do**
3: Randomly choose $x_{\mathcal{C}_j}$ and $y_{\mathcal{C}_j}$ between 0 and 1
4: **end for**
5: **for** a user-defined number of interchanging iterations **do**
6: **for** a user-defined number of model optimizations **do**
7: Choose two random centroids
8: **if** Distance not in distance matrix **then**
9: Calculate the distance between the centroids and store it
10: **end if**
11: Use Equation 2 on these two centroids
12: **end for**
13: **for** a user-defined number of "leaking" opportunities **do**
14: Choose two random centroids $\mathcal{C}_1$ and $\mathcal{C}_2$
15: Choose one item $i \in \mathcal{C}_1$
16: **if** $m\_dist(\{i\}, I_{\mathcal{C}_1} \backslash \{i\}) > m\_dist(\{i\}, I_{\mathcal{C}_2})$ **then**
17: Transfer item i to centroid $\mathcal{C}_2$
18: Adapt the affected distances between the centroids using Equations 3 to 6
19: **end if**
20: **end for**
21: **end for**

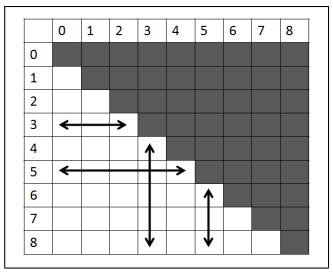


Figure 2: The distance matrix for model with 9 centroids (each white box represents a distance value).

Finally we need to adapt the distance between $\mathcal{C}_i$ and $\mathcal{C}_j$:

$$adaptation = m\_dist(\{a\}, I_{\mathcal{C}_i} \backslash \{a\}) \cdot (|I_{\mathcal{C}_i}| - 1) - m\_dist(\{a\}, I_{\mathcal{C}_j}) \cdot |I_{\mathcal{C}_j}| \tag{5}$$

$$m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_j}) \leftarrow \frac{m\_dist(I_{\mathcal{C}_i}, I_{\mathcal{C}_j}) \cdot |I_{\mathcal{C}_i}| \cdot |I_{\mathcal{C}_j}| + adaptation}{(|I_{\mathcal{C}_i}| - 1) \cdot (|I_{\mathcal{C}_j}| + 1)} \tag{6}$$

Note, in lines 8 and 9 of Algorithm 1, that distances are only calculated when needed. The number of distances to store is a little because of the user-defined number of groups, an advantage of CENTROIDLEAK.

6 Results and Performance

The experiments were done for two main reasons. Push-and-pull depends on the speed of the distance function. For large databases the g_dist function slows down. We want to show that CENTROIDLEAK can better deal with large databases if we can not store the distances between all fragments (items) in the main memory, *scalability*. We made no comparison with the SOM algorithm or any of the traditional clustering algorithms, because the reasons discussed in Section 2 made it less suitable for our purposes. Secondly we want to show that the models made by the algorithm actually approaches the model we expect.

All experiments were performed on an Intel Pentium 4 64-bits 3.2 GHz machine with 3 GB memory. As operating system Debian Linux 64-bits was used with kernel 2.6.8-12-em64t-p4.

The first dataset, called `SyntheticSmall`, is displayed in Figure 3. The dataset consists of 4 groups of 200 two-dimensional items. The second synthetic dataset, called `SyntheticLarge`, is displayed in Figure 5. The dataset consists of 11 groups of 500 two-dimensional items. Note that, contrary to the real-life datasets below, we know the original 2D locations of all data points, from which we compute the distances. The original coordinates are of course not employed by the algorithm, but can be used to verify the performance.

The first real-life dataset we use is the `4069.no_aro` dataset, containing 4,069 molecules; from this we extracted the 1,229 most frequent subgraphs using GSPAN (see [20]). This dataset was provided by Leiden/Amsterdam Center for Drug Research (LACDR). Other datasets we use are datasets of the National Cancer Institute (NCI), and can be found in [13]. One of these datasets contains 32,557 2D structures (molecules, average size is 26.3 nodes) with cancer test data as of August 1999; we will call this dataset the `NCI.CAN.99` dataset.

The approximation of `SyntheticSmall` displayed in Figure 4 shows the groups similarly positioned but slightly turned. The slight turn is caused by the algorithm only knowing the distance between points. Where in traditional push-and-pull you would plot all 800 items in 2D space, now we plot only 100 centroids representing self-optimizing groups of examples. Note that less points make it easier to distinguish (groups of) fragments that are co-occurring. This is essential for any user-interface that allows for further exploration of these centroids (or groups).

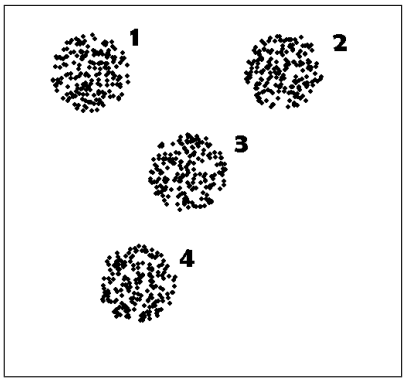


Figure 3: `SyntheticSmall`: Synthetic data when we also know the input vectors, 4 groups of 200 items.

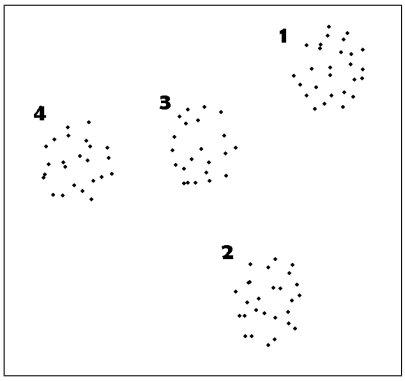


Figure 4: The approximation of `SyntheticSmall` with $n = 100$ centroids ($\alpha = 0.1$, 100 leaks, 10,000 model optimizations and 1,000 iterations).

In Figure 6 the algorithm approaches the `SyntheticLarge` dataset again with only 100 centroids instead of 5,500 (11 groups of 500). Note that we can calculate for the traditional push-and-pull, where we plot all items in 2D space, that $1/2 \cdot n(n-1)$ distances need to be stored and one will eventually run out of memory. Having many more than 5,500 points will eventually be a problem for traditional push-and-pull.

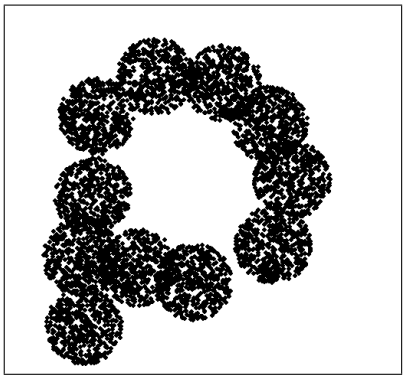


Figure 5: `SyntheticLarge`: Synthetic data when we also know the input vectors, 11 groups of 500 items.

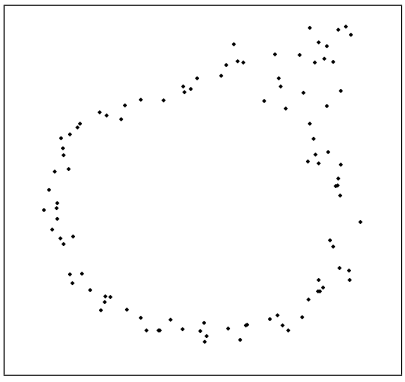


Figure 6: The approximation of `SyntheticLarge` with $n = 100$ centroids ($\alpha = 0.1$, 100 leaks, 10,000 model optimizations and 1,000 iterations).

For the `4069.no_aro` dataset the model of Figure 7 was build. We link those centroids for which items

on average are co-occurring frequently and one can see these are often closer in 2D space.

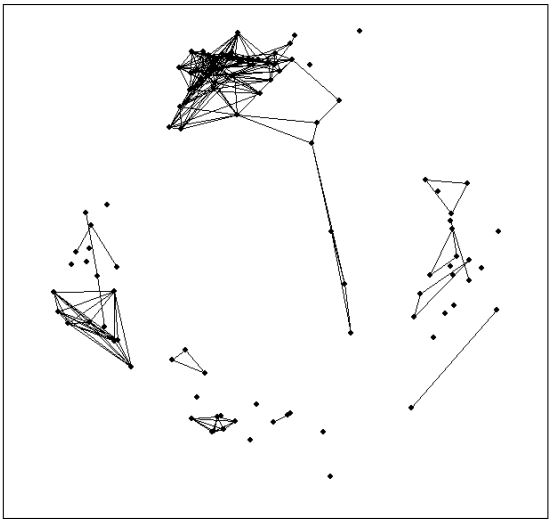


Figure 7: The approximation for the `4069.no_aro` dataset with $n = 100$ centroids ($\alpha = 0.1$, 100 leaks, 10,000 model optimizations, 1,000 iterations).

In Figure 8 it is shown how runtime for a large dataset grows worse for traditional push-and-pull as the number of model optimizations increases. With Figure 8 we want to show that with "leaking" centroids the push-and-pull algorithm becomes more scalable. Due to the grouping of fragments and the lazy updating of the distance matrix there is much less need to count co-occurrence of the fragments.

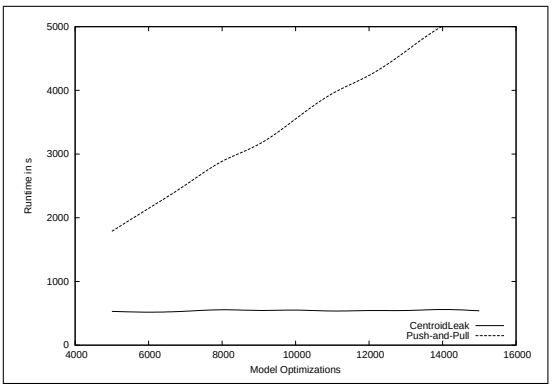


Figure 8: Runtime in seconds with variating model optimizations for the `NCI.CAN.99` dataset with $n = 100$ centroids ($\alpha = 0.1$, 100 leaks, 1,000 iterations).

7 Conclusions and Future Work

In this work we proposed an algorithm for improving the scalability of push-and-pull, a competitive neural network algorithm. The algorithm was able to deal better with larger databases and a great amount of points in the model. This was done by having a fixed amount of centroids where the items are evenly divided among the centroids. Every few iterations certain items have the opportunity to "leak" to a more suitable group. We have experimentally shown that expected models are found. With our algorithm we hope to combine the advantages of Self-Organizing Maps, e.g., one neuron for many similar points, with the advantages of push-and-pull, e.g., good intermediate approximations and visualization with only distance information. The new method gives a better overview, using less points, it only employs intermediate distances, and is easy to understand and to adapt.

In the future we want to make the updating of the distances more lazy, allowing for an even faster algorithm and we want to analyse the biological findings with the improved push-and-pull algorithm.

References

[1] A.M. Bronstein, M.M. Bronstein and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. of National Academy of Sciences (PNAS)* **103** (2006) 1168–1172.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **39** (1977) 1–38.

[3] H. Gao, C. Williams, P. Labute, and J.W. Bajorath. Binary quantitative structure-activity relationship (QSAR) analysis of estrogen. *Chemical Information and Computer Sciences* **39** (1999) 164–168.

[4] P. Gedeck and P. Willet. Visual and computational analysis of structure-activity relationships in high-throughput screening data. *Current Opinion in Chemical Biology* **5** (2001) 389–395.

[5] E.H. de Graaf, J.N. Kok, and W.A. Kosters. Visualization and grouping of graph patters in molecular databases. *Proc. of 27th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2007)*, pp. 267–280 (2007).

[6] J. Hanke, G. Beckmann, P. Bork, and J.G. Reich. Self-organizing hierarchic networks for pattern recognition in protein sequence. *Protein Science Journal* **5** (1996) 72–82.

[7] S. Izrailev and D.K. Agrafiotis. A method for quantifying and visualizing the diversity of QSAR models. *Molecular Graphics and Modelling* **22** (2004) 275–284.

[8] T. Kohonen. *Self Organizing Maps*. Springer Series in Information Sciences, Vol. 30, third extended edition (2001).

[9] W.A. Kosters and M.C. van Wezel. Competitive neural networks for customer choice models. *E-Commerce and Intelligent Methods of Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer*, **105** (2002) 41–60.

[10] E.W. Lameijer, T. Bäck, J.N. Kok, and A.P. IJzerman. Mining a chemical database for fragment co-occurrence: Discovery of "chemical clichés". *Journal of Chemical Information and Modelling* **46** (2006) 553–562.

[11] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proc. of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967).

[12] S. Mahony, D. Hendrix, T.J. Smith, and A. Golden. Self-organizing maps of position weight matrices for motif discovery in biological sequences. *Artificial Intelligence Review Journal* **24** (2005) 397–413.

[13] National Cancer Institute (NCI) website, DTP/2D and 3D structural information, `http://cactus.nci.nih.gov/ncidb2/` [accessed 18.6.2008].

[14] N. Rhodes, P. Willet, J. Dunbar, and C. Humblet. Bit-string methods for selective compound acquisition. *Chemical Information and Computer Sciences* **40** (2000) 210–214.

[15] G. Roberts, G.J. Myatt, W.P. Johnson, K.P. Cross, and P.E. Blower. Leadscope: Software for exploring large sets of screening data. *Chemical Information and Computer Sciences* **40** (2000) 1302–1314.

[16] J.W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* **C-18** (1969) 401–409.

[17] J.B. Tenenbaum, V. de Silva, and J.C. Langford: A global geometric framework for nonlinear dimensionality reduction. *Science* **290(5500)** (2000) 2319–2323

[18] P. Willet, J.M. Barhad, and G.M.J. Downs: Chemical similarity searching. *Journal of Chemical Information and Computer Sciences* **38** (1999) 983–996.

[19] J. Xu, Q. Zhang, and C.-K Shih. V-cluster algorithm: A new algorithm for clustering molecules based upon numeric data. *Molecular Diversity* **10** (2006) 463–478.

[20] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. *Proc. 2002 IEEE International Conference on Data Mining (ICDM)*, pp. 721–724 (2002).

Linguistic Relevance in Modal Logic

Davide Grossi

University of Luxembourg, 6 Rue Coudenhove-Kalergi, Luxembourg

Abstract

The paper shows how a specific notion of (ir)relevance can elegantly be captured in modal logic. More concretely, the paper puts forth a formal definition of linguistic relevance in terms of logical semantics, and provides an axiomatization of it in modal logic.

1 Introduction

The paper presents a short formal study of the notion of linguistic relevance by means of logical semantics and modal logic. By linguistic relevance we mean the possibility of restricting logical evaluation functions and satisfaction relations to subsets of the non-logical language of the original logic. In other words, relevance —as intended here— has to do with the possibility of abstracting from given parts of the non-logical language. For instance, given a propositional logic theory expressed on the alphabet $\{p, q, r\}$, one might want to study the logical consequences of the theory by considering atom r irrelevant and hence abstracting from it. The paper shows how modal logic offers an elegant way to capture this phenomenon without resorting to non-standard tools such as partial evaluation functions.

First, in Section 2, the paper formalizes the notion of linguistic (ir)relevance in logical semantics. Then, in Section 3, the formalization is studied from the point of view of modal logic by first relating it to interesting but not very well-known work done on release logics [3,4], and then by providing a sound and complete axiomatization for it. Some conclusions are drawn in Section 4.

2 "In the beginning was the Word"

The present section introduces the problem of linguistic relevance by illustrating an example first introduced by [2] in the context of Deontic Logic.

2.1 Adam & Eve

Consider the propositional language $\mathcal{L}$ built from the alphabet $\mathbf{P}$ of propositional atoms: `eat_apple` ("the apple has been eaten"), `V` ("a violation has occurred"). We have of course four possible models such that: $w_1 \models \texttt{eat\_apple} \wedge \texttt{V}$, $w_2 \models \texttt{eat\_apple} \wedge \neg\texttt{V}$, $w_3 \models \neg\texttt{eat\_apple} \wedge \texttt{V}$ and $w_4 \models \neg\texttt{eat\_apple} \wedge \neg\texttt{V}$. That is, we have the state in which the apple is eaten and there is a violation (w_1), the state in which the apple is eaten but there is no violation (w_2), the state where the apple is not eaten and there is a violation (w_3), and finally the state where no apple is eaten nor there is a violation (w_4).

Obviously, all these states can be distinguished from each other. But suppose now to compare the models ignoring atom `V`. Models w_1 and w_2 would not be distinguishable any more, nor would states w_3 and w_4. Which is just another way to say that, had we used a sublanguage $\mathcal{L}_i$ of $\mathcal{L}$ containing only atom `eat_apple`, we would have been able to distinguish only states w_1 from w_3 and w_2 from w_4. This latter can be considered to be the language at disposal of Adam & Eve in their pre-moral stage, before hearing God commanding "you shall not eat of the fruit of the tree that is in the middle of the garden"—rather than before actually eating the apple. In fact, after hearing God's command they were already endowed with the possibility to discern good ($\neg$`eat_apple`) from evil (`eat_apple`), that is, their language was enriched and they got to distinguish also states w_1 from w_2 and w_3 from w_4, thanks to the newly introduced notion of violation (`V`).

2.2 Propositional sublanguage equivalence

The intuitions sketched in the previous section are here made formal. Take two propositional models m and m' for a propositional language $\mathcal{L}$. Models w and w' are equivalent if they satisfy the same formulae expressible in $\mathcal{L}$: $w \models \phi$ iff $w' \models \phi$. If w and w' are equivalent ($w \sim w'$) then there is no set Φ of formulae of $\mathcal{L}$ whose models contain w but not w', or vice versa. That is to say, the two models are indistinguishable for $\mathcal{L}$. However, two models which are not equivalent with respect to a given alphabet (a given set of atomic propositions), may become equivalent if only a sub-alphabet (a subset of the atomic propositions) is considered.

Definition 1. *(Propositional sublanguage equivalence) Two models w and w' for a propositional language $\mathcal{L}$ are equivalent w.r.t. sublanguage $\mathcal{L}_i$ if they satisfy the same set of formulae expressible using the alphabet of $\mathcal{L}_i$. For any $\phi \in \mathcal{L}_i$: $w \models \phi$ iff $w' \models \phi$. If w and w' are equivalent w.r.t. $\mathcal{L}_i$ ($w \sim_i w'$) then they cannot be distinguished by any set Φ of formulae of $\mathcal{L}_i$.*

The definition makes precise the idea of two propositional models agreeing up to what is expressible on a given alphabet. To put it another way, it formalizes the idea that two models w and w' are equivalent modulo the alphabet in the complement $-\mathcal{L}_i$ (i.e., $\mathcal{L}\backslash\mathcal{L}_i$) of the sublanguage considered: w is indistinguishable from w' if we disregard the alphabet of $-\mathcal{L}_i$. Notice that if $w \sim_i w'$ and $\mathcal{L}_i = \mathcal{L}$ then $\sim_i = \sim$, that is, $\sim_i$ is the standard equivalence between propositional models.

Proposition 1. *(Properties of $\sim_i$) Let w and w' be two models for the propositional language $\mathcal{L}$. The following holds:*

1. *For every sublanguage $\mathcal{L}_i$ of $\mathcal{L}$, relation $\sim_i$ is an equivalence relation on the set of all models of language $\mathcal{L}$.*
2. *For all sublanguages $\mathcal{L}_i$ and $\mathcal{L}_j$ of $\mathcal{L}$: if $\mathcal{L}_i \subseteq \mathcal{L}_j$ then $\sim_j \subseteq \sim_i$. It follows that for every sublanguage $\mathcal{L}_i$ of $\mathcal{L}$: $\sim \subseteq \sim_i$, that is, standard equivalence implies sublanguage equivalence.*

Proof. Claim (1) is straightforwardly proven. It is easy to see that: identity is a subrelation of $\sim_i$ for any sublanguage $\mathcal{L}_i$; and that $\sim_i \circ \sim_i$ and $\sim_i^{-1}$ are subrelations of $\sim_i$ for any sublanguage $\mathcal{L}_i$. Claim (2) is proven by considering that, if $\mathcal{L}_i$ is a sublanguage of $\mathcal{L}_j$ and $m \sim_j m'$, then for all propositions $\phi \in \mathcal{L}_i$: $w \models \phi$ iff $w' \models \phi$. Hence, $w \sim_i w'$. □

3 Logics for Sublanguage Equivalence Relations

The notion of sublanguage equivalence is here studied from the point of view of modal logic.

3.1 Release logic

Propositional release logics (**PRL**) have been first introduced and studied in [3,4] in order to provide a modal logic characterization of a general notion of irrelevancy. Irrelevancies are, in short, those aspects which we can choose to ignore. Irrelevancy is represented via modal release operators, specifying what is relevant to the current situation and what can instead be ignored. Release operators are indexed by an abstract 'issue' denoting what is considered to be irrelevant for evaluating the formula in the scope of the operator: $\Delta_I\phi$ means 'formula ϕ holds in all states where issue I is irrelevant', or 'ϕ holds in all states modulo issue I' or 'ϕ necessarily holds while releasing issue I'; $\nabla_I\phi$ means 'formula ϕ holds in at least one of the states where issue I is irrelevant', or 'ϕ possibly holds while releasing issue I'.

Issues can be in principle anything, but their essential feature is that they yield equivalence relations which cluster the states in the model. An issue I is conceived as something that determines a partition of the domain in clusters of states which agree on everything but I, or which are equivalent modulo I. Release operators are interpreted on these equivalence relations. As such, propositional release logic can be thought of as a "logic of controlled ignorance" [3]. They represent what we would know, and what we would ignore, by choosing to disregard some issues.

3.1.1 Syntax of PRL

The syntax of **PRL** is the syntax of a standard multi-modal language $\mathcal{L}_n$ [1] where n is the cardinality of the set $\texttt{Iss}$ of releasable issues. The alphabet of $\mathcal{L}_n$ contains: an at most countable set $\mathbf{P}$ of propositional atoms p; the set of boolean connectives $\{\neg, \wedge, \vee, \rightarrow\}$; a finite non-empty set $\texttt{Iss}$ of issues. Metavariables $I, J, ...$ are used for denoting elements of $\texttt{Iss}$. The set of well formed formulae ϕ of $\mathcal{L}^{Prl}$ is defined by the usual BNF:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \Delta_I\phi \mid \nabla_I\phi.$$

where I denotes elements in $\texttt{Iss}$.

One last important feature of **PRL** should be addressed before getting to the semantics. We have seen that modal operators are indexed by an issue denoting what is disregarded when evaluating the formula in the scope of the operator. The finite set $\texttt{Iss}$ of these issues is structured as a partial order, that is to say, $\langle \texttt{Iss}, \trianglelefteqq \rangle$ is a structure on the non-empty set $\texttt{Iss}$, where $\trianglelefteqq$ ("being a sub-issue of") is a binary relation on $\texttt{Iss}$ which is reflexive, transitive and antisymmetric. The aim of the partial order is to induce a structure on the equivalence relations denoting the release of each issue in $\texttt{Iss}$: if $I \trianglelefteqq J$ then the clusters of states obtained by releasing J contain the clusters of states obtained by releasing I. Intuitively, if I is a sub-issue of J then by disregarding J, I is also disregarded. This aspect is made explicit in the models which, for the rest, are just Kripke models.

3.1.2 Semantics of PRL

The semantics of **PRL** is given via the class $\mathfrak{Prl}$ of frames $\mathcal{F} = \langle W, \{R_I\}_{I \in \texttt{Iss}} \rangle$ such that W is a non-empty set of states and $\{R_I\}_{I \in \texttt{Iss}}$ is a family of equivalence relations such that: if $I \trianglelefteqq J$ then $R_I \subseteq R_J$. Models are, as usual, structures $\mathcal{M} = \langle \mathcal{F}, \mathcal{I} \rangle$ where $\mathcal{I}$ is an evaluation function $\mathcal{I} : \mathbf{P} \longrightarrow \mathcal{P}(W)$ associating to each atom the set of states which make it true. **PRL** models are therefore just $\mathbf{S5}_n$ models with the further constraint that the granularity of the equivalence relations follows the partial order defined on the set of issues: the $\trianglelefteqq$-smaller is the issue released, the more granular is the partition obtained via the associated equivalence relation. The satisfaction relation is standard. Boolean clauses are omitted.

Definition 2. *(Satisfaction for* **PRL** *models) Let $\mathcal{M}$ be a* **PRL** *model.*

$$\begin{array}{lll} \mathcal{M}, w \models \Delta_I\phi & \textit{iff} & \forall\, w', wR_Iw' : \mathcal{M}, w' \models \phi \\ \mathcal{M}, w \models \nabla_I\phi & \textit{iff} & \exists w', wR_Iw' : \mathcal{M}, w' \models \phi. \end{array}$$

where $I \in \texttt{Iss}$. As usual, a formula ϕ is said to be valid in a model $\mathcal{M}$, in symbols $\mathcal{M} \models \phi$, iff for all w in W, $\mathcal{M}, w \models \phi$. It is said to be valid in a frame $\mathcal{F} \in \mathfrak{Prl}$ ($\mathcal{F} \models \phi$) if it is valid in all models based on that frame. Finally, it is said to be valid on the class of frames $\mathfrak{Prl}$ ($\mathfrak{Prl} \models \phi$) if it is valid in every frame $\mathcal{F}$ in $\mathfrak{Prl}$.

3.1.3 Axiomatics of PRL

Finally, the axiomatics amounts to a multi-modal **S5** plus the PO (partial order) axiom:

(P)	all tautologies of propositional calculus
(K)	$\Delta_I(\phi_1 \rightarrow \phi_2) \rightarrow (\Delta_I\phi_1 \rightarrow \Delta_I\phi_2)$
(T)	$\Delta_I\phi \rightarrow \phi$
(4)	$\Delta_I\phi \rightarrow \Delta_I\Delta_I\phi$
(5)	$\nabla_I\phi \rightarrow \Delta_I\nabla_I\phi$
(PO)	$\Delta_I\phi \rightarrow \Delta_J\phi$ IF $J \trianglelefteqq I$
(Dual)	$\nabla_I\phi \leftrightarrow \neg\Delta_I\neg\phi$
(MP)	IF $\vdash \phi_1$ AND $\vdash \phi_1 \rightarrow \phi_2$ THEN $\vdash \phi_2$
($\texttt{N}^I$)	IF $\vdash \phi$ THEN $\vdash \Delta_I\phi$

where $I, J \in \texttt{Iss}$. A proof of the soundness and completeness of this axiomatics w.r.t. to the semantics presented in Definition 2 is exposed in [4].

3.1.4 PRL with Boolean Algebras

The partial order structure of a PRL logic is reflected in the axiomatics by axiom P0, and in the semantics by a partial order on the accessibility relations. By adding structure to the partial order on the set of issues more validities can be derived which mirror that structure. Interesting for our purposes is the case when Iss is structured according to a Boolean Algebra. The following propositions lists some of the PRL validities holding in that case.

Proposition 2. *(Validities of PRL with BA) Let* Iss *be ordered as* $\langle \texttt{Iss}, \sqcup, \sqcap, -, 1, 0, \leqq \rangle$*, where the structure* $\langle \texttt{Iss}, \sqcup, \sqcap, -, 1, 0 \rangle$ *is a Boolean Algebra. The following formulae can be derived in PRL:*

$$\Delta_1 \phi \rightarrow \Delta_I \phi \tag{1}$$
$$\Delta_I \phi \rightarrow \Delta_0 \phi \tag{2}$$
$$\Delta_{I \sqcup J} \phi \rightarrow (\Delta_I \phi \wedge \Delta_J \phi) \tag{3}$$
$$\Delta_{I \sqcup J} \phi \rightarrow (\Delta_I \Delta_J \phi \wedge \Delta_J \Delta_I \phi) \tag{4}$$
$$\Delta_I \phi \vee \Delta_J \phi \rightarrow \Delta_{I \sqcap J} \phi \tag{5}$$
$$\Delta_I \phi \leftrightarrow \Delta_{--I} \phi \tag{6}$$
$$(\Delta_I \phi \rightarrow \Delta_J \phi) \leftrightarrow (\Delta_{-J} \phi \rightarrow \Delta_{-I} \phi) \tag{7}$$

Proof. The desired derivations are easily obtainable: some (Formulae 1, 2, 3) are just instances of P0, some (Formulae 5, 6, 7) can be proven by application of P0 and propositional logic. Formula 4 is derived by applying P0, 4 and propositional logic. □

3.2 Sublanguage equivalence in PRL

Logic **PRL** is a viable tool for reasoning about sublanguage equivalence, and thus about the type of linguistic relevance it captures.

3.2.1 Sublanguage equivalence as release

Reasoning about propositional sublanguage equivalence is an instance of reasoning in release logic.

Proposition 3. *(Sublanguage equivalence is a form of release) Consider a propositional language* $\mathcal{L}$ *on the set of atoms* $\mathbf{P}$*, and a non-empty set of states* W*. Any evaluation function* $\mathcal{I} : \mathbf{P} \longrightarrow \mathcal{P}(W)$ *determines a* **PRL** *model* $m = \langle W, \{\sim_{-i}\}_{i \in \mathfrak{Sub}(\mathcal{L})}, \mathcal{I} \rangle$*.*

Proof. It follows from the properties of $\sim_i$ proven in Proposition 1. □

Notice that the release issues Iss are the complements $-\mathcal{L}_i$ of the sublanguages in $\mathfrak{Sub}(\mathcal{L})$. In fact, what is released is just what cannot be expressed. The accessibility relations should therefore be taken to be the sublanguage-equivalence relations $\sim_{-i}$.

Notice also that the set Iss is ordered by set-theoretic inclusion $\subseteq$ between sublanguages of $\mathcal{L}$. In fact, sets of issues have a natural algebraic structure. Let $\mathcal{L}$ be a propositional at most countable language, and let us denote with $\mathcal{L}_i$ any of its sublanguages, i.e., languages defined on a set of atomic propositions $\mathbf{P_i} \subseteq \mathbf{P}$, where $\mathbf{P}$ is the set of atomic propositions. Now let $\mathfrak{Sub}(\mathcal{L})$ be the set of all the sublanguages $\mathcal{L}_i$ of $\mathcal{L}$[1]. If we allow $\mathbf{P_i}$ for any i to be possibly empty, it is immediately clear that the structure $\langle \mathfrak{Sub}(\mathcal{L}), \cup, -, \mathcal{L}, \emptyset \rangle$ is a set algebra and therefore a Boolean Algebra. Leaving technicalities aside, this just means that by choosing an alphabet, a set of sublanguages is consequently chosen which is structured according to a Boolean Algebra. As a consequence, sublanguage equivalence satisfies also the schemata in Proposition 2.

3.2.2 Adam & Eve in PRL models

Intuitively, what Proposition 3 says is that **PRL** is a suitable logic to reason about scenarios like the Adam & Eve one sketched in Section 2.1. Let us get back to that example. Now it is possible to represent both the pre- and post- God's commandment situations, within the same formalism, by making use of the release operators of **PRL**. Suppose Adam & Eve to be at state w_1 in the model with domain $W = \{w_1, w_2, w_3, w_4\}$

[1] Obviously, if $\mathcal{L}$ is infinite then $|\mathfrak{Sub}(\mathcal{L})| > \aleph_0$. For our purposes, we are typically interested in finite languages.

and evaluation $\mathcal{I}$ as in Section 2.1. Recall that the language was built on atoms $\mathbf{P} = \{\texttt{eat\_apple}, \texttt{V}\}$. So let us denote with $\{\texttt{V}\}$ and $\{\texttt{eat\_apple}\}$ the sublanguages containing only atom V and, respectively, atom eat_apple. These sublanguages represent the releasable issues together with the empty language 0 and the full language 1 = **P**. Let $\mathcal{M} = \langle W, \{\sim_{\{\texttt{V}\}}, \sim_{\{\texttt{eat\_apple}\}}, \sim_0, \sim_1\}, \mathcal{I}\rangle$ be the resulting release model. We have that:

$$\mathcal{M}, w_1 \models \texttt{eat\_apple} \wedge \texttt{V} \tag{8}$$

$$\mathcal{M}, w_1 \models \Delta_0(\texttt{eat\_apple} \wedge \texttt{V}) \tag{9}$$

$$\mathcal{M}, w_1 \models \Delta_{\{\texttt{V}\}}\texttt{eat\_apple} \wedge \neg\Delta_{\{\texttt{V}\}}\texttt{V} \tag{10}$$

So Formula 8 just states what holds in w_1, which is the actual state where Adam & Eve eat the apple committing a violation. Formula 9 does the same by saying that, if you evaluate eat_apple and V after releasing nothing, i.e., by using the full descriptive power of the language, then both eat_apple and V necessarily hold. In fact, in the model at issue the set of states reachable from w_1 via $\sim_0$ coincides with w_1 itself, since there are no other states in W which are equivalent with w_1 if all available atoms are used in the comparison. Hence, in the model at issue, Δ_0 refers to the current evaluation state, i.e., w_1. Formula 10 shows what the effects of releasing atom V are. In fact, by abstracting from V, state w_1 is not distinguishable any more from state w_2: $w_1 \sim_{\texttt{V}} w_2$. Hence there exists a state $w_2 \in W$ such that $\mathcal{M}, w_2 \models \texttt{eat\_apple} \wedge \neg\texttt{V}$.

Formulae 10 and 9 represent Adam & Eve's situation after and, respectively, before God's commandment "you shall not eat of the fruit of the tree that is in the middle of the garden". Such commandment introduces a further characterization of reality, exemplified here by the notion of violation, which was not available to Adam & Eve before the commandment was uttered.

3.3 Linguistic release logic: PRL$^{\mathcal{L}}$

Although all sublanguage equivalence relations are **PRL** accessibility relations (Proposition 3), it is easy to see that the reverse does not hold. In a sense the characterization of sublanguage equivalence in terms of **PRL** is too liberal. This section proposes a logic specifically tailored for talking about sublanguage equivalence w.r.t. a propositional language $\mathcal{L}$: linguistic release logic (**PRL**$^{\mathcal{L}}$ in short).

3.3.1 Syntax of PRL$^{\mathcal{L}}$

Linguistic release logic makes use of the same multi-modal language $\mathcal{L}_n$ of **PRL**. The important difference is that the set of issues $\texttt{Iss} \subseteq \mathcal{P}(\mathcal{L})$, that is, the issues are sublanguages $\mathcal{L}_i$ of a given propositional language $\mathcal{L}$. To avoid clutter in the notation, modal operators will be indexed with $i, j, \ldots$ instead of $\mathcal{L}_i, \mathcal{L}_j, \ldots$.

3.3.2 Semantics of PRL$^{\mathcal{L}}$

Let us first precisely define the notion of sublanguage equivalence model, which was somehow already sketched in Proposition 3.

Definition 3. *(se-models) A sublanguage equivalence model (se-model in short) $m = \langle W, \{\sim_{-i}\}_{i \in \mathfrak{Sub}(\mathcal{L})}, \mathcal{I}\rangle$ is a* **PRL** *model where each $\sim_{-i}$ is a sublanguage equivalence relation w.r.t the sublanguage $\mathcal{L}_{-i}$ of a given propositional language $\mathcal{L}$, that is, it is a* **PRL** *model where the release accessibility relation $\sim_{-i}$ obeys the constraint: $w \sim_{-i} w'$ iff $\forall p \in \mathcal{L}_{-i}$: $w \models p$ iff $w' \models p$.*

In other words, se-models are those **PRL** models where the accessibility relations and the evaluation function are related in such a way that two states can access one another iff they satisfy the same propositional atoms. Therefore, the set of se-models built on a $\mathfrak{Prl}$ frame is a subset of **PRL** models built on that frame.

The satisfaction relation for these models is defined as follows. Boolean clauses are omitted.

Definition 4. *(Satisfaction for se-models) Let $\mathcal{M}$ be a se-model.*

$$\mathcal{M}, w \models \Delta_i\phi \quad \textit{iff} \quad \forall\, w', w \sim_{-i} w' : \mathcal{M}, w' \models \phi$$

$$\mathcal{M}, w \models \nabla_i\phi \quad \textit{iff} \quad \exists w', w \sim_{-i} w' : \mathcal{M}, w' \models \phi.$$

where $i \in \texttt{Iss} = \mathfrak{Sub}(\mathcal{L})$. As usual, a formula ϕ is said to be valid in a se-model $\mathcal{M}$, in symbols $\mathcal{M} \models \phi$, iff for all w in W, $\mathcal{M}, w \models \phi$. It is said to be valid in a frame $\mathcal{F} \in \mathfrak{Prl}$ ($\mathcal{F} \models \phi$) if it is valid in all se-models based on that frame. Finally, it is said to be valid on the class of frames $\mathfrak{Prl}$ ($\mathfrak{Prl} \models \phi$) if it is valid in every frame $\mathcal{F}$ in $\mathfrak{Prl}$.

Intuitively, $\Delta_i\phi$ means that ϕ holds in all states that are equivalent to the evaluation state up to sublanguage $-\mathcal{L}_i$ (or, that can be reached by releasing sublanguage $\mathcal{L}_i$), and $\nabla_i\phi$ means that ϕ holds in at least one state which is equivalent to the evaluation state up to sublanguage $-\mathcal{L}_i$ (or, that can be reached by releasing sublanguage $\mathcal{L}_i$).

3.3.3 Axiomatics of PRL$^{\mathcal{L}}$

The point is to find, given a language $\mathcal{L}$ an axiomatics which is sound and complete w.r.t. the class of se-models built on the class of $\mathfrak{Prl}$ frames. The axiomatics extends the axiomatics of **PRL** with axiom `NoCross`:

(P)	all tautologies of propositional calculus
(K)	$\Delta_i(\phi_1 \to \phi_2) \to (\Delta_i\phi_1 \to \Delta_i\phi_2)$
(T)	$\Delta_i\phi \to \phi$
(4)	$\Delta_i\phi \to \Delta_i\Delta_I\phi$
(5)	$\nabla_i\phi \to \Delta_i\nabla_i\phi$
(PO)	$\Delta_i\phi \to \Delta_j\phi$ IF $j \leqq i$
(NoCross)	$\Delta_i p \vee \Delta_i\neg p$ IF $p \notin i$
(Dual)	$\nabla_i\phi \leftrightarrow \neg\Delta_i\neg\phi$
(MP)	IF $\vdash \phi_1$ AND $\vdash \phi_1 \to \phi_2$ THEN $\vdash \phi_2$
(N^I)	IF $\vdash \phi$ THEN $\vdash \Delta_i\phi$

where $i, j \in$ `Iss`. The idea behind the new axiom is that, if an atom p does not belong to a sublanguage $i \in \mathfrak{Sub}(\mathcal{L})$ (i.e., if p belongs to the complement of i), then all worlds which are accessible by releasing sublanguage i must either satisfy p or $\neg p$. Technically, `NoCross` forces the release accessibility relation not to cross the bipartitions of the domain W yielded by each atom p, when p does not belong to the released language. This could be considered to be the distinctive feature of linguistic release with respect to other forms of release[2]. A proof of the soundness and completeness of this axiomatics w.r.t. the class of se-models built on $\mathfrak{Prl}$ frames (Definition 4) is provided in Appendix A.

4 Conclusions and Future Work

The contribution of the paper consisted in the formalization of linguistic (ir)relevance in terms of propositional sublanguage equivalence (Section 2), its study as an instance of propositional release logic (Section 3.2), and its axiomatization in modal logic (**PRL**$^{\mathcal{L}}$) which we called linguistic release logic (Section 3.3).

A future research line which is worth pursuing is the study of linguistic release logic in combination with other modal operators, in particular epistemic and doxastic ones. This could clarify interesting epistemic phenomena such as the so-called "unknown-unknowns"[3]. Getting back to the Adam & Eve example, and stretching it a little bit, we can see that before God's commandment it was an "unknown-unknown" for both Adam and Eve that eating the apple counted as a sin. In other words, neither did they know that eating the apple was a sin, nor did they know that they did not know that eating the apple was a sin. We think that logic **PRL**$^{\mathcal{L}}$ in combination with **S5** could provide interesting insights on this issue.

Acknowledgments

This study was supported by the *Ministère de la Culture, de L'Enseignement Supérieur et de la Recherche, Grand-Duché de Luxembourg* (grant BFR07/123).

[2] The reader is referred to [3, 4]

[3] We thank Martin Caminada for pointing us to this notion. "Unknown unknowns" have obviously to do with the failure of negative introspection.

References

[1] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001.

[2] Davide Grossi. Pushing Anderson's envelope: The modal logic of ascription. In R. van der Meyden and L. van der Torre, editors, *Proceedings of the 9th International Conference on Deontic Logic in Computer Science (DEON 2008), Luxembourg, Luxembourg, July 15-18, 2008th International Workshop on Deontic Logic in Computer Science (DEON 2008), Luxembourg, Luxembourg, July 15-18, 2008*, number 5076/2008 in LNAI, pages 263–277. Springer, 2008.

[3] J. Krabbendam and J.-J. Ch. Meyer. Contextual deontic logics. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 347–362, Amsterdam, 2003. IOS Press.

[4] J. Krabbendam and J.-J.Ch. Meyer. Release logics for temporalizing dynamic logic, orthogonalising modal logics. In M. Barringer, M. Fisher, D. Gabbay, and G. Gough, editors, *Advances in Temporal Logic*, pages 21–45. Kluwer Academic Publisher, 2000.

A Linguistic Release Logic is Sound and Complete

The proof of soundness is routinary. It is well-known that inference rules `MP` and `N` preserve validity on any class of frames, and that axioms `T`, `4` and `5` are valid on models built on equivalence relations[4]. Providing the soundness of $\mathbf{PRL}^{\mathcal{L}}$ w.r.t. the class of se-models built on $\mathfrak{Prl}$ frames boils then down to checking the validity of axioms `PO` and `NoCross`.

Theorem 1. *(Soundness of* $\mathbf{PRL}^{\mathcal{L}}$ *w.r.t* $\mathfrak{Prl}^{\mathcal{L}}$ *models) Logic* $\mathbf{PRL}^{\mathcal{L}}$ *is sound w.r.t. the class* $\mathfrak{Prl}^{\mathcal{L}}$ *of models, i.e., if* $\vdash_{\mathbf{PRL}^{\mathcal{L}}} \phi$ *then* $\models_{\mathfrak{Prl}^{\mathcal{L}}} \phi$.

Proof. Validity of `PO`. Suppose axiom `PO` is not valid. This means that $\exists \mathcal{M} \in \mathfrak{Prl}^{\mathcal{L}}$ s.t. for at least one world w, it holds that $j \leqq i$ and $\mathcal{M}, w \models \Delta_i \phi \wedge \neg \Delta_j \phi$, that is: $\forall w'$ such that $w \sim_{-i} w'$ $\mathcal{M}, w' \models \phi$ but $\exists w$" s.t. $w \sim_{-j} w'$ and $\mathcal{M}, w'' \not\models \phi$. This contradicts Proposition 1 since $\mathcal{L}_{-j} \subseteq \mathcal{L}_{-i}$. Validity of `NoCross`. If $p \in i$, then $p \in -i$, hence $\forall w, w'$ s.t. $w \sim_{-i} w'$ we have that $\mathcal{M}, w' \models p$ iff $\mathcal{M}, w \models p$ by Definition 1. It follows that $\forall w$ either $\mathcal{M}, w \models \Delta p$ or $\mathcal{M}, w \models \Delta \neg p$. □

The proof of completeness makes use of the standard canonical model technique. Notice that we have to deal with completeness w.r.t. to the class of se-models which can be built on the class $\mathfrak{Prl}$ of propositional release logic frames.

Lemma 1. *(Redifining strong completeness for* $\mathbf{PRL}^{\mathcal{L}}$*) Logic* $\mathbf{PRL}^{\mathcal{L}}$ *is strongly complete w.r.t. the class of frames* $\mathfrak{Prl}$ *iff every* $\mathbf{PRL}^{\mathcal{L}}$*-consistent set* Φ *of formulae is satisfiable on some se-model built on a frame in class* $\mathfrak{Prl}$.

Proof. From right to left we argue by contraposition. If $\mathbf{PRL}^{\mathcal{L}}$ is not strongly complete w.r.t. the class $\mathfrak{Prl}^{\mathcal{L}}$ then there exists a set of formulae $\Phi \cup \{\phi\}$ s.t. $\Phi \models_{\mathfrak{Prl}} \phi$ and $\Phi \nvdash_{\mathbf{PRL}^{\mathcal{L}}} \phi$. It follows that $\Phi \cup \{\neg\phi\}$ is $\mathbf{PRL}^{\mathcal{L}}$-consistent but not satisfiable on any sublanguage equivalent model built on a frame in class $\mathfrak{Prl}$. From left to right we argue per absurdum. Let us assume that $\Phi \cup \{\neg\phi\}$ is $\mathbf{PRL}^{\mathcal{L}}$-consistent but not satisfiable in any sublanguage equivalent model built on a frame in class $\mathfrak{Prl}^{\mathcal{L}}$. It follows that $\Phi \models_{\mathfrak{Prl}} \phi$ and hence $\Phi \cup \{\neg\phi\}$ is not $\mathbf{PRL}^{\mathcal{L}}$-consistent, which is impossible. □

Now let $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}}$ be the canonical model of logic $\mathbf{PRL}^{\mathcal{L}}$ in the multi-modal language $\mathcal{L}_n$. Model $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}}$ is the structure $\left\langle W^{\mathbf{PRL}^{\mathcal{L}}}, \{R_i^{\mathbf{PRL}^{\mathcal{L}}}\}_{1 \leq i \leq n}, \mathcal{I}^{\mathbf{PRL}^{\mathcal{L}}} \right\rangle$ where:

1. The set $W^{\mathbf{PRL}^{\mathcal{L}}}$ is the set of all maximal $\mathbf{PRL}^{\mathcal{L}}$-consistent sets.
2. The canonical relations $\{R_i^{\mathbf{PRL}^{\mathcal{L}}}\}_{1 \leq i \leq n}$ are defined as follows: for all $w, w' \in W^{\mathbf{PRL}^{\mathcal{L}}}$, if for all formulae ϕ, $\phi \in w'$ implies $\nabla_i \phi \in w$, then $w R_i^{\mathbf{PRL}^{\mathcal{L}}} w'$.
3. The canonical interpretation $\mathcal{I}^{\mathbf{PRL}^{\mathcal{L}}}$ is defined by $\mathcal{I}^{\mathbf{PRL}^{\mathcal{L}}}(p) = \{w \in W^{\mathbf{PRL}^{\mathcal{L}}} \mid p \in w\}$.

[4] See [1].

As usual, we have to prove the Existence and Truth Lemmata for logic $\mathbf{PRL}^{\mathcal{L}}$.

Lemma 2. *(Existence lemma) For all states in $W^{\mathbf{PRL}^{\mathcal{L}}}$, if $\nabla_i\phi \in w$ then there exists a state $w' \in W^{\mathbf{PRL}^{\mathcal{L}}}$ s.t. $R_i^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$ and $\phi \in w'$.*

Proof. The claim is proven by construction. Assume $\nabla_i\phi \in w$ and let $w'_0 = \{\phi\} \cup \{\psi \mid \Delta_i\psi \in w\}$. Set w'_0 must be consistent since otherwise there would exists $\psi_1, \ldots, \psi_m \in w'_0$ such that $\vdash_{\mathbf{PRL}^{\mathcal{L}}} (\psi_1 \wedge \ldots \wedge \psi_m) \rightarrow \neg\phi$, from which we obtain $\vdash_{\mathbf{PRL}^{\mathcal{L}}} (\Delta_i\psi_1 \wedge \ldots \wedge \Delta_i\psi_m) \rightarrow \Delta_i\neg\phi$. Since $\Delta_i\psi_1, \ldots, \Delta_i\psi_m \in w$ we have that $\neg\nabla_i\phi \in w$, which contradicts our assumption. Therefore, w'_0 is consistent and can be extended to a maximal $\mathbf{PRL}^{\mathcal{L}}$-consistent set (for Lindenbaum's Lemma[5]). By construction, w' contains ϕ and is such that for all ψ, if $\Delta_i\psi \in w$ then w' contains ψ. From this it follows $R_i^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$ since, if this was not the case, then there would exist a formula ψ' s.t. $\psi' \in w'$ and $\nabla_i\psi' \notin w$. Since w is maximal $\mathbf{PRL}^{\mathcal{L}}$-consistent, $\Delta_i\neg\psi' \in w$ and hence $\neg\psi' \in w'$, which contradicts the $\mathbf{PRL}^{\mathcal{L}}$-consistency of w'. □

Lemma 3. *(Truth lemma) For any formula ϕ: $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}}, w \models \phi$ iff $\phi \in w$.*

Proof. The claim is proven by induction on the complexity of ϕ. The Boolean case follows by the properties of maximal $\mathbf{PRL}^{\mathcal{L}}$-consistent sets. As to the modal case, it follows from the definition of the canonical relations $R_i^{\mathbf{PRL}^{\mathcal{L}}}$ and Lemma 2. □

Everything is now put into place to prove the strong completeness of $\mathbf{PRL}^{\mathcal{L}}$.

Theorem 2. *(Completeness of $\mathbf{PRL}^{\mathcal{L}}$ w.r.t $\mathfrak{Prl}$) Logic $\mathbf{PRL}^{\mathcal{L}}$ is strongly complete w.r.t. the class $\mathfrak{Prl}$ of frames, i.e., if $\Phi \models_{\mathfrak{Prl}} \phi$ then $\Phi \vdash_{\mathbf{PRL}^{\mathcal{L}}} \phi$.*

Proof. By Proposition 1, given a $\mathbf{PRL}^{\mathcal{L}}$-consistent set Φ of formulae, it suffices to find a model state pair $(\mathcal{M}, w)$ such that: (a) $\mathcal{M}, w \models \Phi$, (b) $\mathcal{M}$ is an se-model. Let $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}} = \left\langle W^{\mathbf{PRL}^{\mathcal{L}}}, \{R_i^{\mathbf{PRL}^{\mathcal{L}}}\}_{1\leq i\leq n}, \mathcal{I}^{\mathbf{PRL}^{\mathcal{L}}}\right\rangle$ be the canonical model of $\mathbf{PRL}^{\mathcal{L}}$, and let Φ^+ be any maximal consistent set in $W^{\mathbf{PRL}^{\mathcal{L}}}$ extending Φ. By Lemma 3 it follows that $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}}, \Phi^+ \models \Phi$, which proves (a). To prove (b), we show that $\mathcal{M}^{\mathbf{PRL}^{\mathcal{L}}}$ is s.t.: (b.1) the frame $\mathcal{F}$ on which $\mathcal{M}$ is based is a $\mathfrak{Prl}$ frame; and (b.2) $R_i^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$ iff $\forall p \in \mathcal{L}_{-i}$: $p \in w$ iff $p \in w'$. As to (b.1), it is well-known that axioms `T`, `4` and `5` force the relations $R_i^{\mathbf{PRL}^{\mathcal{L}}}$ to be equivalence relations. It remains to be shown that if $i \leqq j$ then $R_{-j}^{\mathbf{PRL}^{\mathcal{L}}} \subseteq R_{-i}^{\mathbf{PRL}^{\mathcal{L}}}$. Assume $R_{-j}^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$ and $\Delta_j\phi \in w$. It follows that $\phi \in w'$ and, for axiom `PO`, that $\Delta_i\phi \in w$ and, for axiom `T` that $\nabla_i\phi \in w$. Therefore, $R_{-i}^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$.

As to (b.2), form left to right. Assume $R_{-i}^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$. For any $p \in \mathcal{L}_{-i}$, if $p \in w$, then for axiom `NoCross` $p \in w'$. If $p \notin w$ then, for axiom `NoCross` $p \notin w'$. From right to left. Assume that for all $p \in \mathcal{L}_{-i}$, $p \in w$ iff $p \in w'$. Suppose that $p \in w'$. For the maximal $\mathbf{PRL}^{\mathcal{L}}$-consistency of w it holds that $\nabla_i p \in w$. Hence, if $p \in w'$ then $\nabla_i p \in w$, and therefore $R_{-i}^{\mathbf{PRL}^{\mathcal{L}}}(w, w')$. This completes the proof. □

[5] See [1]

Beating Cheating: Dealing with Collusion in the Non-Iterated Prisoner's Dilemma

N. Höning [a] T. Kozelek [b] M.C. Schut [a]

[a] *VU University, Amsterdam,* {nhg400,mc.schut}@few.vu.nl
[b] *Charles University, Prague*, tomas.kozelek@gmail.com

Abstract

The Iterated Prisoner's Dilemma (IPD) is a well-known challenging problem for researching multi-agent interactions in competitive and cooperative situations. In this paper, we present the Ask-First (*AF*) strategy for playing multi-agent non-Iterated PD (nIPD) that is based on evolving trust chains between agents. Each agent maintains a (relatively small) table containing trust values of other agents. When agents are to play each other, they ask their neighbours what trust they put in the opponent. Chains are then followed until an agent is found that knows the opponent and the trust value is propagated back through the chain. The played move is then decided based upon this trust value. When two agents have played each other, they update their trust tables on the basis of the outcome of the game. The strategy is first evaluated in a benchmark scenario where it is shown that it outperforms a number of benchmark strategies. Secondly, we evaluate the strategy in a scenario with a group of colluding agents. The experiments show that the *AF* strategy is successful here as well. We conclude that the *AF* strategy is a highly flexible, scalable and distributed way (the chain topology adapts to the way that agents are picked to play each other) to deal with a difficult multi-agent nIPD problem (i.e., robust against collusions).

1 Introduction

In open e-commerce systems (e.g., an electronic market), *collusion* is a serious threat to the honest operation of the system. Collusion is the agreement between two or more persons to deceive others to obtain an secretive (or: illegal) objective. On eBay, this happens by so-called *shil bidding*, where friends of the seller bid on an item solely with the intention to raise the price. These friends can even be other eBay accounts of the seller himself and not necessarily real persons. Another example of collusion happens when in recommender systems, items are evaluated positively by the seller's friends (which may, again, be fake accounts of the seller). In such situations, it is very important for a potential buyer (who is truthful) to judge the reliability of the bids or the recommendations. Ideally, one wants to achieve this without having to rely on a trusted third-party or authority.

In this paper, we suggest a way to do exactly this based on the idea of *trust networks*. In such networks, agents maintain the trust that they have in each other and communicate this with each other when necessary. The intuition behind this method is that in real life, we ask our friends how they feel about a certain item or some particular shop. This intuition has lead us to develop the very simple Ask-First (*AF*) strategy that faithful agents can employ to protect them from malicious or unreliable agents.

Within the context of Iterated Prisoner's Dilemma (IPD), the strategy works as follow (we provide more detailed information later in the paper). Firstly, an *AF* strategy agent maintains a (relatively small) table containing trust values of other agents. Then, when agents are to play each other, they ask their neighbours what trust they put in the opponent. Next, chains are then followed until an agent is found that knows the opponent and the trust value is propagated back through the chain. Then, the played move is then decided based upon this trust value. Finally, when two agents have played each other, they update their trust tables on the basis of the outcome of the game.

The research objective of this paper is two-fold: 1) to show that the *AF* agents outperform the benchmark agents (all-cooperators and all-defectors); and 2) to show that the *AF* agents outperform collusion agents. For objective 2), we also want to show that the *AF* agents use the created network chains effectively and that colluding agents cannot invade these chains.

This paper has the following structure. In Section 2, we present the background for our work: computational trust and reputation, referral networks and social networks. We also explain the non-Iterated Prisoner's Dilemma. Section 3 lays out the details of our framework and the Ask-First (*AF*) strategy. In Section 4 we show by a number of experiments the effectiveness of the *AF* strategy. Finally, in Section 5, we draw conclusions about the performed study and provide some pointers for future work.

2 Background

The work presented in this paper, builds further on research on computational trust and reputation, referral networks, and social networks. In this paper, we place all within the context of the Iterated Prisoner's Dilemma (IPD). In this Section, we briefly touch upon each of the topics.

2.1 Trust, Reputation and Referral

The concept of *trust* is essential in societies and open systems in order to maintain 'good' social interactions and commitments between individuals. As mentioned above, on eBay, you want to trust the person from whom you are buying your items. In real life, we have many (albeit implicit, unconscious) mechanisms operating for managing trust; in virtual life (internet), we have not yet established such mechanisms [1]. Much research in the last decade has been dedicated to the question how to use trust as a mechanism for regulating social interaction. In the early 1990s, Marsh [7] presented a formalisation of trust that pins down a number of defining properties of trust in order to facilitate a precise discussion about trust. The formalism was implemented for a multi-agent system in a PD scenario, demonstrating a recognisable behaviour of trust among the agents.

An often-used mechanism for managing trust is by means of *reputation*: a societal indication of how much you can trust someone, reflecting its past actions. Like the concept of 'trust', 'reputation' is a convoluted term and needs to be clarified unambiguously for precise discussion and application; Mui *et. al* [9] do this by giving a concise overview of the notion within the context of multi-agent systems. In terms of applications, current major websites (eBay, Amazon) have *centralised* reputation mechanisms in place, where people's ratings about each other are collected, processed and communicated back. However, such centralisation of reputation is not always possible, e.g., for peer-to-peer service provision [14]. It can be expected that future systems will become increasingly more open and require such decentralised mechanisms for maintaining trust and reputation.

A recent development in the search for "decentralised reputation management" concerns so-called *referral networks*. In such networks, agents communicate information about trust and reputations are built based on this information. The networks are used for query-based searching for information and expertise in a person's social network [5]. In a referral network, nodes have neighbours (whom they query directly) and acquaintances (whom they query only when referred to); both sets are dynamic (neighbours can become acquaintences and vice versa) and usually limited. Studies of referral networks by Singh *et. al* [14] have looked at how network structure evolved under various circumstances; how agent learning models (enabling agents to learn about each other in terms of expertise – producing correct answers, and sociability – providing good referrals) affects the quality of the work; and how to design self-organising referral networks.

2.2 Social Networks

The basis of the referral networks described above is the *social network* that connects individuals within a collective system. In such a network of individuals, there are links between these individuals that represent, for example, friendship, kinship or values. Social network analysis views these networks as graphs, in which the nodes are individuals and the edges are the links. Within the context of this paper, research on social networks includes, for example, the study of decentralised search algorithms [6] and investigation of the relationship between social network topologies and emergent behaviour [2].

A particular stream of research worth mentioning is the investigation of so-called *small-world networks*. Since the formalisation of the small world problem (originally coined by Milgram [8]) by Watts and Strogatz [13] in the nineties, much research on social networks looks at such networks being small-world networks: it takes relatively few steps to go from one random node to another. This type of networks has two important properties: a short average path length (hence, the relatively few steps to reach other nodes), and a high

clustering coefficient (number of a node's neighbours that also know each other). The experiments presented later in this paper also consider small-world networks.

2.3 Non-Iterated Prisoner's Dilemma

The well-known Iterated Prisoner's Dilemma (IPD) is a generic abstract representation of complex social dilemmas [3]. It is a (non-zero-sum) game where two players make decisions simultaneously whether to defect or cooperate and receive rewards based on the combination of their two actions. The ordinance of the payoffs makes this game a dilemma: [dc] > [cc] > [dd] > [cd], where $[xy]$ denotes the reward for player 1 where player 1 plays x, player 2 plays y and $x, y \in \{$c(ooperate), d(efect)$\}$. The dominant strategy for the one-shot PD (where the game is played once) is for both players to defect. The iterated version of the game (where it is played for a number of times) does not have a single dominant strategy. In a competition-based evaluation of the IPD, it was found that an extremely simple strategy beats the other strategy: *tit-for-tat*, where the opponent's last action was simply copied (with default action 'cooperate').

In this paper, we have a population of agents that repeatedly plays pairwise PD games. Note, however, that it is not an implementation of IPD, because the agents change their interaction partner in every round and have a limited memory of their previous opponents. Like in [4, 12], we call this version the *non-Iterated PD* (nIPD).

While in the one-shot and iterated PD it is almost certain that the players will ultimately defect each other, in the nIPD this is not the case at all. Much research has been done on the *evolution of cooperation* in nIPD, addressing the important question how cooperation can arise between selfish agents. Nowak and May [11] showed that in a two-dimensional cellular automata (where a cell could be in a 'cooperate' or 'defect' state) if the PD was used as the update rule, patterns of cooperation and defection emerged. In follow-up work by Olifant [12], it is shown that in a society of agents, *spatiality* plays an important role: agents that are close by each other are more likely to play than those that are further away. This work gives rise to the issue of how agents are connected to each other: can the structure of the connection network affect the evolution of cooperation?

In more recent work, Ellis and Yao [4] address this issue by looking at a *social network* inspired approach for the evolution of cooperation. In their (non-spatial) nIPD, links are formed between cooperating agents. These links are reinforced by repeated cooperation, while defection breaks a link. All links taken together represent the social network of the agents. Ellis and Yao present a strategy that can exploit this network: the *discriminator* agent bases its move (cooperate or defect) on the *centrality* of its opponent. (This centrality represents the agent's reputation among its neighbours.) The experiments for this strategy showed that the discriminator strategy is more successful than all-cooperators or all-defectors. In another series of experiments, agents are able to evolve strategies while playing – ranging from all-cooperate to all-defect. The evolving parameters included 1) the probability that an agent interacts with another agent that has a lower reputation than some threshold value, and 2) this threshold. The experiments showed that when agents are able to observe the centrality, then the population evolves to all-cooperators.

3 The *Ask-First* Strategy

The main idea of the *Ask-First* strategy is to *ask* around before you *act*. For the nIPD, this means asking a neighbour for advise about whether to (c)ooperate with or (d)efect an unknown opponent. The asking process is recursive: when you are asked for advise about someone you do not know either, you forward the question to a neighbour of your own, and so forth. This asking around leads to so-called *information chains* between agents over which trust information is communicated. Note that the recursive nature of these information chains in our approach sets it apart from referral networks (which were explained above), because these work *iteratively*.

The protocol for each agent employing the *AF* strategy[1] involves the following steps: select best neighbour, ask this neighbour for advise, process the advise, play the PD game and do some aftermath (updating). We describe each of these steps in the following Sections. Before this, we first explain the dynamics of the social network of the agents.

[1] In terms of reciprocity, the reputation-based *AF* strategy is a *downward* kind of indirect reciprocity [10, p.1292]: individual a has helped b and *therefore* receives help from c.

Network Dynamics We are interested in testing the *AF* strategy in a set of agents that are connected with each other through a small-world network. In other words, this paper is not about showing that the evolved network is small-world; that is our starting point and we show that the *AF* agents can successfully exploit such networks. However, we do not want to simply impose the small-world property on the network; instead, we select PD-playing couples in such a way that a small-world social network evolves. As mentioned above, small-world networks have low average path lengths L (comparable to L of a random graph of the same size) and a high clustering coefficient C (much higher than C of a random graph). For building small-world social networks, one could follow Watts and Strogatz [13]: often connect neighbouring agents and occasionally remote agents. Here, we use a more refined method that employs a concept of *distance*, based on an approach suggested by Kleinberg [6].

Let us define the concept of distance as follows. Firstly, let each agent x be uniquely identified by an integer $I_x \in [1, N]$, where N is the total number of agents. We assume that agents whose identifiers are close to each other are more likely to interact with each other. For example, in the context of eBay agents, this could express some common interests in goods. In the context of our nIPD agents, it simply means that close agents are more likely to play each other. Let a *distance* measure $dist(a, b)$ represent how close agent a is to agent b; in general: $dist(x, y) = min(abs(I_x - I_y), N - abs(I_x - I_y))$, where x, y are agents and I_x, I_y their identifiers, respectively. (Note that this function makes agents I_1 and I_N close to each other.)

For building small-world networks, agents are connected with each other if there is little distance between them. In other words. If a, b and c are agents, then a should have a higher probability to be connected with b then with c if $dist(a, b) < dist(a, c)$. Formally, let x, y be agents and the probability that they are connected be $\frac{1}{dist(x,y)^\alpha}$ (according to Kleinberg), where α is some given constant; then small-world networks emerge when α is around 2. With these notions, we use Kleinberg's probability-vector approach [6] to evolve the desired small-world social networks. For the experiments presented below, we have obtained empirical evidence supporting this. We have not included these measurement for reasons of space.

3.1 Select Best Neighbour

Let an agent a have a *neighbourhood* Ω, containing the names of at most K other agents. For each of the agents in the neighbourhood, agent a has assigned a *trust value* representing how much a trusts this agent: let this trust value be $\tau \in \mathbb{R}$, where $\tau \in [0, 1]$ (representing minimum to maximum trust, respectively). If agent a plays a neighbour and this neighbour defects, then its trust decreases and vice versa (later we explain the calculation of this value in detail).

When faced with an opponent o that is not in the agent's neighbourhood, the agent chooses one of its neighbours, let this be agent b, to ask for advise about playing o. The choice for a specific neighbour is based on 1) how much agent a trusts agent b, and 2) what the distance between agents b and z is. We represent the trust of agent a in agent b by $trust(a, b)$; the distance between them by $dist(a, b)$. The two measurements are integrated into the function $eval(a, b, o)$ that expresses the evaluation of neighbour b by agent a about opponent o. In general, $eval(a, b, o) = \omega_{trust} * trust(a, b) + \omega_{distance} * (1/dist(b, o))$, where $a, b \in \Omega$, $o \notin \Omega$, $\omega \in [0, 1]$ and $\sum_\omega = 1$. This function thus *weighs* trust and distance: the higher the weight of distance, the more probable it is that the opponent is reached soon; a high weight of trust may produce more reliable chains but it is also more likely to fail in finding the opponent at all. Finally, an agent decides to ask the neighbour whom it evaluates highest: $max_{b \in \Omega}\ eval(a, b, o)$.

3.2 Ask for Advise

After agent a has decided to ask neighbour b about opponent o, he thus asks agent b for advise. Asking is recursive, thus when agent b does not have o in his Ω, then he selects his best neighbour to ask for advise about o. If it is not possible to find an agent that knows the opponent within 6 steps[2], then the chain is not built and agent a uses his default behaviour (here: cooperate).

If a successful chain was built (i.e., o was found within 6 steps), then the trust value is propagated back to agent a, multiplied by the trust values in the information chain I: $\Pi_{a \in I} trust(a, bn(a))$, where $bn(a)$ is the best neighbour of a. Consider, for example, that the built chain goes from a via b to agent c who knows o. Then the propagated trust value received by a is $trust(c, o) \times trust(b, c) \times trust(a, b)$.

[2]The number '6' was chosen because of the infamous "six degrees of separation" [8].

3.3 Process Advise

If agent a has the propagated trust value t, then it can still decide whether to use the received advise or not. This decision is based on a system-wide *chain trust* threshold θ_{chain} (here: 0.3). If $t > \theta_{chain}$, then a uses the advise; otherwise not (and a uses his default behaviour – cooperate). After that, a's decision whether to cooperate or not[3] is based on the (also system-wide) threshold θ_{trust} (here: 0.5). If $t > \theta_{trust}$, then a cooperates; otherwise he defects.

3.4 Play and Aftermath

Playing the game is straightforward: each player makes his move and receives a payoff based according to a given nIPD payoff table. If an agent has finished a game, three things happen. Firstly, the opponent is added to the neighbourhood (if he was not in there yet). Secondly, the trust value of the opponent is updated according to the following rule[4]:

$$\textbf{if } (cooperation) \textbf{ then } \tau_{new} = \tau_{old} + (1 - \tau_{old})/2$$

$$\textbf{if } (defection) \textbf{ then } \tau_{new} = \tau_{old} - (\tau_{old})/2,$$

where $\tau_{old} = 0.5$ iff the opponent was not yet a neighbour. Note that these function are both sigmoid curves. This means that if an opponent changes his behaviour, it will have a quick effect on the trust in him. This follows recommendations by Axelrod [3] in that successful strategies should quickly retaliate and forgive. The opponent is then added to Ω with the Kleinberg probability[5] $1/dist(a, o)^{\alpha}$. If the opponent is added and the neighbourhood size exceeds the allowed K entries, then a random agent is removed from Ω (in order to keep the size within K).

4 Experimental Evaluation

We present an experimental evaluation of the effectiveness of the *AF* strategy, in which we compare the performance of agents employing the *AF* strategy with *colluding* agents (who do not reveal true trust values). As mentioned, all the experiments involve a nIPD scenario, in which agents repeatedly play PD games with other agents. Each experiment consists of a fixed number of iterations; in each iteration, an opponent is chosen once for each agent, after which both players remain in the pool. This means that each agent plays at least one game in each iteration, and probably more (in which case the payoffs within one iteration are accumulated). Every third iteration, an *evolutionary update* happens (like in [4]): two random agents are selected, of which the one with lowest average payoff[6] is replaced by a new agent that follows the strategy of the other agent (removed agents are also removed from all neighbourhoods). In this Section, we subsequently present 1) the research hypotheses, 2) the strategies of the agents, 3) the experimental design and setup, and 4) the results and analysis.

4.1 Research Hypotheses

The objectives stated in the introduction translate into the following 3 hypotheses:

- **Hypothesis 1:** The AF strategy is successful against defection.
- **Hypothesis 2:** The AF strategy is successful against collusion.
- **Hypothesis 3:** Colluding agents cannot spread false information over the information chains.

We measure the successfulness of a strategy by means of the proportion of agents that employ that strategy. Additionally, we measure *Utilitarian Social Welfare*, which is the sum of payoffs all agents achieved in the current iteration. For hypothesis 3, we look at the information chains and compare the number of *built* chains with the number of *used* chains and the number of *only-AF* chains (those chains that only contain *AF* agents).

[3] If agent a has opponent o in his own neighbourhood, then he uses this exact same decision procedure (where he knows t himself).

[4] Note that the only trust value to be updated is thus the opponent's one, not the trust value of the best neighbour.

[5] Earlier we explained that based on this probability, it is decided which agents are playing each other. Here, the constant α is relatively high to support locality.

[6] This is the total payoff accumulated over its lifetime.

4.2 Strategies

We employ some simple strategies to evaluate their interplay. If not mentioned otherwise, all agents can be asked to build up information chains and report their true trust evaluations about agents they know.

Always Cooperate (AC) – This strategy cooperates in every interaction.

Always Defect (AD) – This strategy always defects the opponent.

Ask First (AF) – This is the strategy that was explained in detail in Section 3.

Simple Collusion (SC) – This strategy behaves like AD, i.e., it always defects. However, when this agent, say m, is asked for the trust value of a neighbour n, it returns $1 - trust(m, n)$. The consequence of this is that SC agents give each other good ratings and cooperating agents bad ratings.

4.3 Design and Setup

We considered three different experiments, between which we varied the initial populations: in experiment 1, the initial population consists $\frac{1}{2}$ AD agents and $\frac{1}{2}$ AF agents; in experiment 2, it consists of $\frac{1}{3}$ AD agents, $\frac{1}{3}$ AC agents and $\frac{1}{3}$ AF agents; in experiment 3, it consists of $\frac{1}{2}$ AF agents and $\frac{1}{2}$ SC agents. In all experiments, there were 20 runs for each experiment; the population size (N) was 150; the number of iterations was 600 (for exp.2, we also report on an experiment with 3,000 iterations). Regarding the neighbourhoodsize and the weights, we tested each with two values and use the average of the obtained results for the analysis: we let $K \in [0.7 * logN * \sqrt{logN}, 1.1 * logN * \sqrt{logN}]$, and $\omega_{distance} \in [0.3, 0.7]$.

4.4 Results and Analysis

Figures 1–3 show the results of experiments 1, 2 and 3, respectively. This section reviews our research hypotheses.

Hypothesis 1 – For experiment 1, this hypothesis holds. We conducted a Welsh Two-Sample T-Test ($t = 51.1728, df = 712.562, p\text{-}value < 2.2e - 16$) to test the difference between the population proportions of both strategies at iteration 600 (the mean population size of AF was 183.99 and the mean population size of AD was 28.41). Figure 1(a) shows the strategy proportions over all iterations. It is noteworthy to say that if a cooperating strategy like AF takes over the population, this is not only self-beneficiary. Figure 1(b) shows the Utilitarian Social Welfare that is maximised as AF agents take over the population. The high spike in the beginning results from the initially trusting behaviour of AF agents facing unknown AD agents. Quickly, AF agents learn not to trust AD agents and spread that information via information chains to other AF agents.

For experiment 2, this hypothesis does not hold: the AF strategy loses from the AD strategy (see Figure 2(a)). We conducted another Welsh Two-Sample T-Test ($t = 28.7613, df = 1024.961, p\text{-}value < 2.2e - 16$), showing that the AD agents significantly outnumber the AF agents (the mean population sizes were 63.47 and 142.78 for AF and AD, respectively). However, Figure 2(b) shows that the AF strategy recovers once there are no pure cooperators left.

Hypothesis 2 – This hypothesis holds. We conducted a Welsh Two-Sample T-Test ($t = 67.04, df = 669, p\text{-}value < 2.2e - 16$) to test on the difference of the population representation of both strategies in the end of runs in experiment 3 (AF, SC). The results show with strong significance that the AF strategy outnumber the SC strategy after $N * 4$ iterations (see Figure 3(a)). The mean of AF was 205.26 and the mean of SC was 19.74.

Hypothesis 3 – To understand the setting of AF agents against SC agents better, we monitored the use of information chains. In particular, we protocolled the number of chains that were successfully built (i.e. the original asker, an AF agent, got an answer of someone who had his opponent in his trust table), the number of information chains that were actually used (i.e. the returning trust information exceeded the chain trust threshold) and the number of the used chains which only contained AF agents. Figure 3(b) shows that AF agents will use exclusively use chains that only consists of other AF agents. This may seem strange at first, but it is a consequence of the design. Defecting agents will seldom be chosen for building the next step in a chain as they quickly rank low in trust at all AF agents, due to the retaliating nature of the AF strategy. If they are chosen, they lower the overall returned trust, making it highly unlikely that the asking AF agent will

use the chain for his decision. In this sense, the AF strategy could be called unforgiving: defecting agents will not be asked for information[7].

5 Conclusions

We proposed and implemented a distributed reputation system, that effectively exploits a small-world social network. The introduced *Ask-First* strategy employed by the agents is based on ask-first-act-later: if faced with an unknown opponent, these agents ask around their social network, and decide on dealing with the opponent based on the advise received from this network. By means of a series of experiments, we showed that our strategy can handle malicious agents that do not reveal honest advise when they are asked for. We tested the strategies in a multi-agent non-Iterated Prisoner Dilemma scenario, but expect that it can be useful for a number of practical applications, e.g., distributed web service provision or for distributed trust management in open recommender systems (eBay or Amazon). In future work, we will go more into putting the strategy into practice.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the Hawaii International Conference on System Sciences*, 2000.

[2] G. Abramson and M. Kuperman. Social games in a social network. *Phys. Rev. E*, 63(3), 2001.

[3] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[4] T.S. Ellis and X. Yao. Evolving cooperation in the non-iterated prisoner's dilemma: a social network inspired approach. In *IEEE Congress on Evolutionary Computation*, 2007.

[5] Henry A. Kautz, Bart Selman, and Mehul A. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[6] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians*, 2006.

[7] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science, University of Stirling, 1994.

[8] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[9] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 280–287. ACM Press, 2002.

[10] A. Nowak and K. Sigmund. Evolution of indirect reciprocity. *Nature*, 437:1291–1298, 2005.

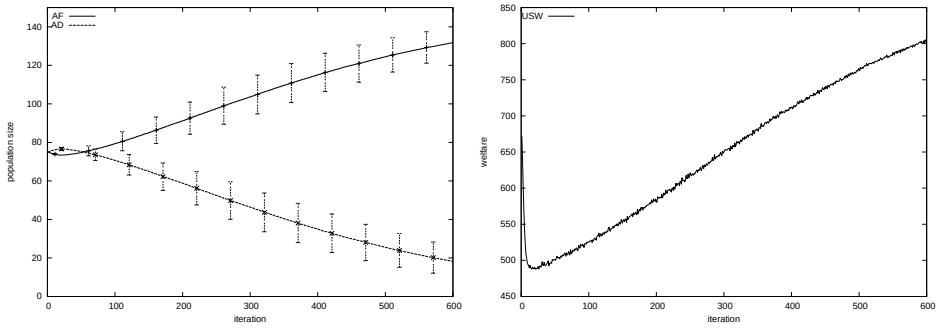
[11] M.A. Nowak and R.M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.

[12] M. Oliphant. Evolving cooperation in the non-iterated prisoner's dilemma: The importance of spatial organization. In R. Brooks and P. Maes, editors, *Proceedings of Artificial Life IV*, pages 349–352. MIT Press, 1998.

[13] J. Duncan Watts and H. Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393, June 1998.

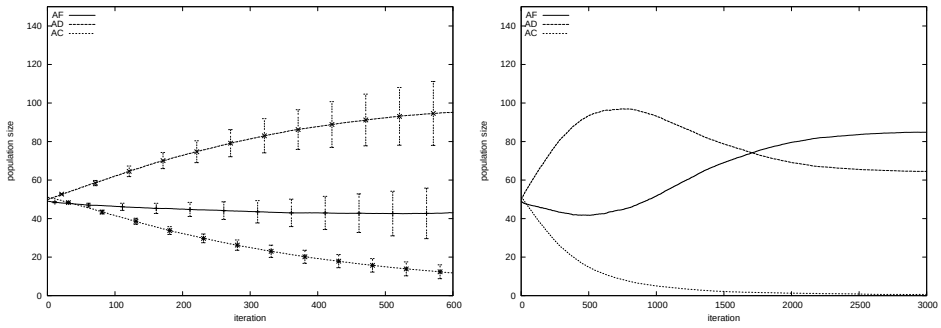
[14] B. Yu, M.P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Proceedings of the First IEEE Symposium on Multi-Agent Security and Survivability*, pages 1–10, 2004.

[7] In some follow-up experiments (not reported here), we tested the forgiveness of the *AF*-strategy by putting them in a world with *alternating* agents (who change from always-cooperate to always-defect agents every x iterations. The results of these experiments showed that *AF* agents respond quite rapidly to such changes.



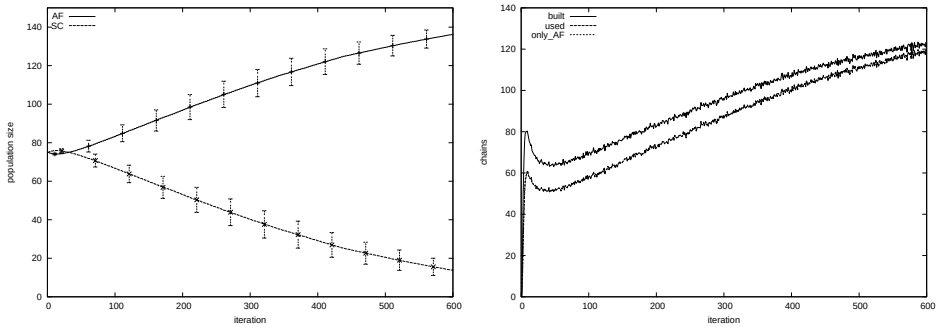
(a) Strategy populations (b) Utilitarian Social Welfare

Figure 1: AD vs AF in experiment 1, N=150



(a) Strategy populations (b) Strategy populations (3,000 iterations)

Figure 2: AD vs AC vs AF in experiment 2, N=150



(a) Strategy populations (b) Information chain usage

Figure 3: AF vs SC in experiment 3, N=150

The Influence of Physical Appearance on a Fair Share

Steven de Jong [a] Rob van de Ven [a] Karl Tuyls [b]

[a] *Maastricht University, P.O.Box 616, 6200 MD Maastricht*
[b] *Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven*

Abstract

There are many factors that influence people's concept of a fair share. Personal preferences, additional information, as well as previous experience make humans choose different strategies when interacting with different opponents under otherwise similar circumstances. To account for such factors computationally, we previously introduced a utility-based model called *priority awareness*. In this paper, we present a study with human participants that provides additional evidence that humans use priorities when they need to think about a fair share. We perform this study by using a method called *human computation*, which can be used to explicitize implicit information by asking many people to perform only a few tasks. More specifically, we look at the impact of physical appearance on what humans consider a fair share. In an online survey, we present people with a few fictive opponents represented by a photo, and ask them to play an Ultimatum Game with these opponents. We aggregate the human data into a ranking and find that appearance has a strong influence on what people consider a fair share, i.e., people associate a different priority with different opponents, purely based on immediate appearance. The same method may be used to elicit other implicit influences on human priorities. Such influences may then be taken into account by developers of, for instance, multi-agent systems.

1 Introduction

Allocating resources to a number of agents is a common but challenging problem, since any division needs to balance efficiency and fairness. Welfare economics proposes a set of axioms and collective utility functions that may address this problem [4, 10]. Experiments with humans reveal that humans may have completely different measures of what constitutes a satisfactory allocation [15, 26]. In summary, the human concept of a fair share is influenced by efficiency considerations (i.e., individual rationality), but also by factors such as personal preferences (i.e., some people are more greedy than others) [15, 16], additional information (i.e., if we know our opponent to be poor, we may be more willing to share) [8] and previous experience (i.e., if we played against our opponent before, we may expect him to behave in a similar manner this time) [2, 12, 25]. Eliciting such factors is a challenging task, since very often, a carefully designed experiment needs to be conducted with many participants that are not (yet) acquainted. Often, researchers need to motivate participants by giving them a monetary reward that is somehow related to their performance in the experiment at hand. In the end, obtained data may still be biased or difficult to interpret.

Human computation [30] provides an interesting alternative to such expensive lab studies. Originally, it was proposed as an alternative to developing complicated algorithms. The task at hand (e.g., tagging images to elicit the objects present in them) is usually presented as a game to attract voluntary participants. Each participant then has to perform a small, carefully selected subtask that is simple for humans, yet complicated for computers (e.g., labelling the objects seen in a few images). The results of the performed subtasks are then aggregated (e.g., into a database with suitable tags for many images).

In this paper, we show that the usefulness of human computation is not restricted to providing alternatives for complicated algorithms; it may also be successfully applied to elicit implicit factors that influence human decision-making. More precisely, we investigate whether physical appearance triggers different behavior with respect to a fair share, by letting our participants play the Ultimatum Game [17]. In this game, the first player proposes how to divide a sum of money with the second player. If the second player rejects this

division, neither gets anything. If the second accepts, the first gets his demand and the second gets the rest. The individually rational solution is for the first player to propose the lowest possible amount, and for the second player to accept. Human players however consistently offer more, and reject low offers [15]. In earlier work [8], we observed that additional (explicit) information on the wealth of opponents significantly influences participants' offers. This behavior has been described in a model called priority awareness [7]. In the current work, the participants in our survey also obtain additional information concerning their (fictive) opponents in the Ultimatum Game, but this time the information is implicit and represented by means of a photograph. The photos may trigger previously stored stereotypes [9, 31] concerning, e.g., gender, race, age, and (apparent) wealth. Such stereotypes are inherently implicit and therefore difficult to make explicit.

Using our survey, we aim at eliciting implicit information such as stereotypes. To this end, participants have to answer five questions each. In every question, we present them with two opponents that are selected in a way that ensures the most (expected) information gain. We then ask which of the two would receive more money (where it is also possible to give the same amount to both). Clearly, such qualitative questions concerning ordinal preferences pose a simpler task to the participants than having to answer many quantitative questions related to cardinal preferences, such as "how much would you give to this person?". We only ask some of these quantitative questions to determine whether our participants' answers align with results reported in literature [e.g., 8, 26]. Using the (relative) answers obtained from our participants, we construct a ranking of our set of photos, from the least-earning person to the most-earning person. This requires additional computation to translate participants' preferences concerning pairs of photos to a global ranking concerning the whole set of photos. Recent research has proposed a method called CollaboRank to perform this task [21].

Thus, our contribution in this paper is two-fold. First, we provide additional support for the fact that humans actively use priorities when they are deciding upon sharing with others. This information may be explicit, but may also be implicit. Especially in the latter case, eliciting information is a difficult task that typically requires many well-controlled experiments. This leads to the second contribution: we show how human computation methods, such as collaborative ranking, may be used to elicit the information that human decisions are based on. This elicitation is relevant, not only for psychologists, but also for developers of multi-agent systems, since many multi-agent systems are interacting with humans [27]. Moreover, human strategies are often objectively better than known strategies based on, e.g., individual rationality [1, 6, 7].

The remainder of this paper is structured as follows. In Section 2, we briefly look at background information, discussing existing work aimed at explaining human decision-making, as well as the human-computation algorithms applied in our work. In Section 3, we present our methodology, which consists of a survey and an analysis of the data provided by our participants. In Section 4, we present the results of our analysis, aimed at assessing the influence of appearance on human behavior. Finally, in Section 5, we conclude and look at future work.

2 Background

In this section, we briefly discuss work aimed at explaining and modelling human decision-making. Moreover, we discuss the CollaboRank method used both to make participants' input as useful as possible as well as to construct our ranking.

2.1 Explaining human decision-making

In contrast to what was believed in the 1940s and 1950s when game theory was gaining attention, humans are not perfectly individually rational [16, 22]. Instead of purely focussing on the optimization of their own reward, humans also care about how this reward compares to the reward of others. Given a different opponent, a human player may choose a completely different strategy, given that all other circumstances are kept identical. This behavior is motivated by a large number of factors, some related to the player, some to the opponent. Many of these factors have been identified and modelled. Here, we provide only a small overview. For details, we refer to [7].

Inequity aversion. In [15], this is defined as follows: *"Inequity aversion means that people resist inequitable outcomes; i.e., they are willing to give up some material payoff to move in the direction of more equitable outcomes"*. To model inequity aversion, an extension of the classical game theoretic actor is introduced, named Homo Egualis [15, 16]. The Homo Egualis utility function basically leads to agents not experiencing their own private payoff x, but also a penalty for other agents doing better, and a smaller

penalty for other agents doing worse. It has been shown to adequately describe human behavior in various games, including the Ultimatum Game [15] and the Public Goods Game [6]. In both cases, human strategies may actually lead to better payoffs than strategies that are generally considered in the multi-agent systems community, i.e., individually rational strategies [7].

Reciprocal fairness. There are also experiments in which human behavior is not adequately captured by a utility model that is exclusively based on inequity aversion and material interest [3]. The most important limitation of the inequity-averse model is that it does not explicitly explain how fair behavior evolves with repeated interactions between agents [15]. For instance, a group of people repeatedly playing the same game may start by playing in an individually rational manner, but may end up playing in a fair, cooperative manner. Reciprocal fairness models aim at providing an answer to the questions why and how this happens. The main idea is that humans cooperate because of direct and indirect reciprocity – here, direct means that a person is nice to someone else because he[1] expects something in return from this other person, and indirect means that an agent is nice to someone else because he expects to obtain something from a third person. It turns out that the opposite, i.e., punishing someone who is nasty, has an even greater effect on cooperation [29]. However, being nasty may be costly, and thus, it would be individually rational to punish only when we are sure to encounter the object of punishment again. Once again, humans do not select the individually rational solution: even in one-shot interactions, they consistently apply punishment if this is allowed. Since this is clearly not of direct benefit to the punisher, this phenomenon is referred to as altruistic punishment (see, e.g., [13, 14, 32]). The question thus seems to shift from 'why do people cooperate?' to 'why do people perform costly punishment?'. Various explanations have been analyzed from the perspective of evolutionary game theory [16]. For instance, many researchers argue that altruistic punishment only pays off when the reputation of the players becomes known to everyone [12, 25]. There are also alternative explanations such as volunteering [18, 19], fair intentions [11] or the topology of the network of interaction [28].

Priority awareness. In [8], the influence of additional information on the human concept of fairness (or a fair share) is studied. Experiments with human subjects show that additional information matters strongly, even in the absence of previous interactions. This phenomenon is not adequately captured in existing models. We proposed a model called *priority awareness* that is able to address additional information [7]. The priority-awareness model has already been applied to Ultimatum Games in which participants were told that their opponent was ten times poorer or ten times wealthier than them; we also varied the amount at stake between $\$10$ and $\$100$K. Clearly, participants' priorities relative to their opponents change as a result of this explicit information. Poor opponents are given substantially more money; an increasing amount at stake reduces the amount offered and accepted. This phenomenon is not described in most existing studies [20, 26]. In this paper, we extend our previous work on priority awareness. We determine whether people's priorities also change due to implicit information, which is much harder to explicitize. We use the approach of human computation to investigate this.

2.2 Human computation and ranking

The CollaboRank method [21] is inspired by Von Ahn's work on the power of human computation [30]. The method has been proposed to perform image-based ranking tasks, e.g., for sorting a set of images on the size of the objects presented in these images. Humans are more capable of recognizing (sizes of) objects in images than computers, and therefore also better at ranking those images. The method may also be used to elicit truly implicit information in the images, as in this paper. Moreover, it is also applicable to other ranking tasks than only images.

The method enables multiple participants to rank a large set of images in a collaborative, iterative way. To this end, it distributes relatively simple ranking tasks, concerning only a small subset of the set of images that need to be ranked, effectively over the participants. The subsets' rankings are aggregated to a global ranking concerning the entire set. Through an iterative process the global ranking should converge to a stable, high quality ranking, unless there is absolutely no consensus concerning the rankings of subsets.

The key element of the CollaboRank method is the Global Preference Matrix (GPM). Below, we will first explain this matrix. Next, we discuss how the matrix is used to create tasks concerning a subset of images, and how results concerning such a subset are integrated in a global ranking.

[1] For brevity, we use 'he' and 'him' wherever 'he or she' and 'him or her' is meant.

2.2.1 The Global Preference Matrix (GPM)

The GPM fullfills two tasks: first, it forms the basis for formulating the tasks given to participants, and second, it aggregates the personal rankings that the participants have submitted into a global ranking $\mathcal{R}$. The preferences of all the possible combinations of image pairs $\langle i, j \rangle$ for which $i, j \in I'$ (the set of images) are stored in the GPM. The preference relation, as represented by the GPM, on image set I' of size ϕ, is represented by a $\phi \times \phi$ preference matrix $\mathbf{P} = (p_{ij})$. A preference $p_{ij} = \frac{1}{2}$ indicates an equal preference for image i and image j (i.e., $i \sim j$). $p_{ij} > \frac{1}{2}$ indicates that i is preferred to j (i.e., $i \succ j$) while $p_{ij} < \frac{1}{2}$ indicates the reverse. In other words, the pair-wise preference p_{ij} is interpreted as the probability of image i being ranked before image j. A property that should permanently hold is $\forall i, j \in \mathbf{P}, p_{ij} = 1 - p_{ji}$.

2.2.2 Formulating a task

The formulation of a task for each participant is the first step of the COLLABORANK method. Because each participant only ranks a small subset of the images, the algorithm should intelligently select the image subsets (in our case, pairs) that will be given to each participants. This minimizes the number of participants needed to arrive at a sufficient global ranking. COLLABORANK uses an entropy function, which ensures that image pairs with the highest level of uncertainty are picked first to be ranked. The uncertainty of the rank probability p_{ij} is calculated using the binary entropy function [24]:

$$H_2(p_{ij}) = -p_{ij} \log_2 p_{ij} - p_{ji} \log_2 p_{ji} \tag{1}$$

The total entropy of an image i is defined as follows:

$$H(i) = \sum_{j \in I : j \neq i} H_2(p_{ij}) \tag{2}$$

This function indicates the uncertainty of the ranked position of an image i. COLLABORANK therefore selects an image m having the highest entropy H. Then, it selects an image k which has the highest entropy as a pair with image m, i.e., $H_2(p_{km})$. The two images m and k are shown as a pair to a participant. Note that this method leads to all possible image pairs being ranked initially, as unranked pairs have a positive entropy, whereas pairs that are ranked by a single participant have an entropy $H = 0$. Only after all pairs have been ranked for the first time does the algorithm select pairs for a second ranking.

After the participant submits his personal ranking, it is aggregated into the GPM. The preferences are updated of those pairs of images in the GPM that were presented to the participants. In this way, the GPM is filled with values that are closer to either 0 or 1, to eventually create a global ranking. More precisely, a personal ranking is aggregated into the GPM by assigning the average preference of images i and j, with regard to all the submitted personal rankings, to p^g_{ij}. For example, when 8 out of 10 participants submit $(i \succ j)$, then $p^g_{ij} = 0.8$ and $p^g_{ji} = 0.2$.

2.2.3 Producing a global ranking

The Greedy-Order algorithm [5] is being used to extract a global ranking from the GPM. To apply this algorithm, the GPM is interpreted as a directed weighted graph, where initially, the set of vertices V is equal to the set of images I', and each edge $u \rightarrow v$ has weight p_{uv}. Each vertex $v \in V$ is assigned a potential value $\pi(v)$, which is the weighted sum of the outgoing edges minus the weighted sum of the ingoing edges:

$$\pi(v) = \sum_{u \in V} p_{vu} - \sum_{u \in V} p_{uv} \tag{3}$$

The vertex t with the maximum potential is selected. The corresponding image, is assigned the rank $R(t) = |V|$, which makes it the first image in the ranking. The vertex and its edges are deleted from the graph, after which the potential value π of the remaining vertices are updated. This process is repeated until the graph is empty. The last vertex removed from the graph has rank 1, and will be the last image in the ranking.

3 Methodology

In order to determine the effect of appearance on the human concept of a fair share, we developed an online survey in which participants are confronted with a small number of opponents, represented by a

You have to divide 10 dollars between you and an opponent. If the opponent accepts your offer, you can keep the money according to this offer. If the opponent rejects the offer, you both receive nothing.

Who would you offer more money?

Opponent A. **Opponent B.**

( ) Opponent A.
( ) Opponent B.
( ) I would offer them an equal amount of money.

Figure 1: A survey question aimed at deriving a global ranking from pairwise comparisons.

photo. The questions of this survey are adapted by the COLLABORANK algorithm in such a way that the information gain per submitted survey is largest, as described above. Ranking is used to aggregate individual participants' answers concerning this small number of opponents into a global result concerning the whole set of photos used. In this section, we take a closer look at the survey.

Control questions. The survey starts with a small number of 'control questions' that were asked to determine whether our participants have strategies similar to those reported in earlier work [see, e.g., 8, 26]. Thus, we asked how much money participants would offer to an unknown opponent. We also varied the amount of money at stake between 10 and $10,000$ dollars, to verify our earlier results [8].

Pairwise ranking. The main body of survey questions (i.e., five questions per participant) concern pairwise comparison of two opponents, which are pulled from a set of 24 photographs. One of the 252 possible photograph pairs (and the associated question) is given in Figure 1.

For the selection of photographs, we used a basis of five properties: gender, age, race, attractiveness, and wealth. We used Google's image search (`http://images.google.com`) to find many photographs that displayed a large diversity in at least one of these properties. The resulting large set of photographs was then reduced to 24 manually, with the aim of keeping sufficient diversity with respect to all properties.

Due to the COLLABORANK method, pairs are not presented at random, but in a way that provides optimal information. For the participants, pairs of photographs should therefore become increasingly difficult to judge, that is, after every pair has been presented to a participant at least once. Since every participant ranks five pairs out of the 252 possible ones, we need 51 participants to have every pair ranked at least once. From the 51^{st} participant onwards, we expect more and more participants to select the last answer option, i.e., 'I would offer them an equal amount of money.'

Quality of the ranking. To address the quality of the pairwise ranking, we also ask participants to quantify how much would be offered to one certain opponent, with a photograph being pulled from the set of 24 photographs randomly (i.e., the photograph shown to a participant for this question may or may not be shown in the pairwise questions). This is clearly a much harder question than having to choose between two opponents. We can create a quantitative ranking by simply sorting the photographs increasingly on average amount offered. If the pairwise and quantitative rankings are sufficiently similar, this tells us that human preferences may indeed be elicited by (simple) qualitative questions instead of (more difficult) quantitative questions. To measure similarity, we use the Kendall rank correlation coefficicient τ [23] to determine the

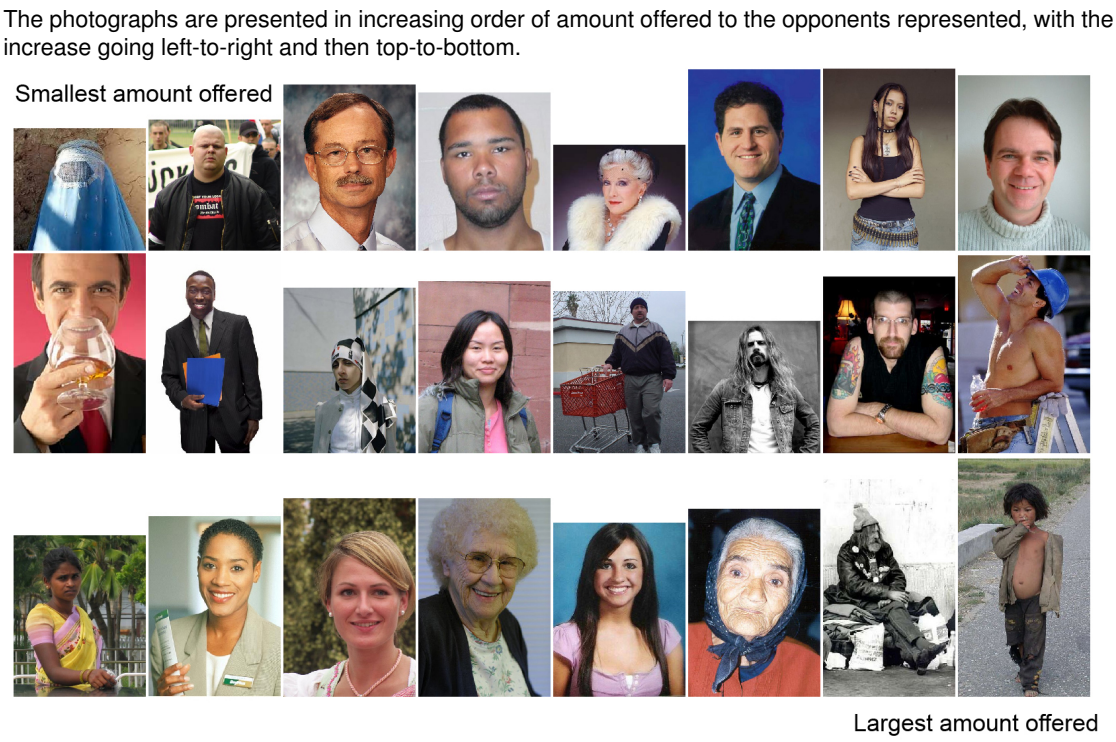


Figure 2: The global relative ranking of 24 visible opponents.

similarity of the rankings. $\tau = +1$ indicates that the two rankings are equivalent; $\tau = -1$ indicates that they are reversed, and $\tau = 0$ indicates they are completely uncorrelated.

4 Results

In this section, we discuss results, obtained from 173 participants that were attracted through promotion on various social websites. First, we look at the control questions, and then, we discuss the various rankings.

Control questions. As mentioned above, the survey starts with a small number of questions aimed at determining whether our participants have strategies similar to those reported in earlier work. When being confronted with random, invisible opponents, most participants offer as well as accept 50% of an amount of 10 dollars, as also reported in literature [8, 26]. Increasing the amount of money to $10,000$ dollars, offering as well as accepting 50% is still the most often-chosen strategy (i.e., it is chosen by $\sim 30\%$ of the participants), but a majority of participants would offer somewhat less money, and would accept substantially less (15% already accept only 10% of the amount, i.e., $1,000$ dollars). This conforms to what we found in our earlier survey [8].

Pairwise ranking. The result of aggregating the 173 pairwise rankings, as submitted by our participants, into a global ranking of the 24 photographs, is given in Figure 2. As the survey was in progress, we noticed that the photograph pairs presented to the participants became increasingly difficult to distinguish, i.e., the option 'I would offer them an equal amount of money.' was chosen more and more often. This indicates that the ranking process is working. Moreover, since 173 participants completed the survey, each performing 5 pairwise rankings, we obtained 865 pairwise rankings. There are 252 different photograph pairs; therefore, if pairs were selected at random, every pair would be ranked a little more than 3 times. Due to COLLABORANK however, we are able to ensure that difficult pairs are ranked more often than easy pairs; some pairs were ranked by more than 15 participants.

Looking at the ranking produced, we see a number of interesting phenomena. For instance, wealth is clearly important: the photographs that seemingly depict poor people are almost all given relatively large amounts (and the reverse). This corresponds to what we found in our earlier survey when we explicitly mentioned the wealth of opponents [8]. Discarding the most-receiving (poorest) people, there also seems

to be rather clear division concerning gender; all remaining photographs on the bottom row (i.e., the most-receiving people) are women, whereas seven out of the ten least-receiving people are men. Concerning attractiveness, we can at least note that facial expression seems important – the least-receiving person has no visible facial expression, and many of the people that receive little money are not smiling. Race does not seem to be a decisive factor for our participants, as the various skin colours are not clearly grouped. The same goes for age, if we discard people that are stereotypically wealthy or poor.

Quality of the ranking. Since we asked participants one quantitative question, we can expect every photograph to be quantitatively ranked an expected $173/24 = 7.2$ times. Due to this low number, some of the photographs displayed no significant difference concerning quantitative information. Other photographs yield very clear results; for instance, the least-receiving photo in the qualitative ranking also receives the least in the quantitative ranking (i.e., 2.2 ± 1.6 out of 10). Two of the three most-receiving photographs in the qualitative ranking are also in the top-three of the quantitative ranking, with offers as high as 5.6 ± 2.2 out of 10, i.e., significantly more than the least-receiving photograph and also more than 50% of the total amount. This would not be possible if subjects were only motivated by inequity aversion [8, 15]. Comparing the two rankings analytically, we observe that the Kendall rank correlation coefficient $\tau = 0.51$, which indicates a significant correspondance. Thus, humans display roughly the same preferences, based on physical appearance stereotypes, in both a pairwise comparison as well as in more difficult quantitative questions.

5 Conclusion

Previously, we introduced a model called priority awareness [7] that not only explains how priorities may influence human decisions, but also allows agents in a multi-agent system to display human-like behavior with respect to priorities. The model has already been supported by experiments with human participants in which additional explicit information concerning the opponents was given.

In this paper, we support the model further by experiments in which implicit information is given, i.e., partipants are able to see their opponents on photographs, which may trigger one or more stereotypes. To elicit implicit factors present in the photographs, we use a method called human computation [21, 30], which allows researchers to present participants with small tasks that are simple for them, yet would be very difficult for a computer. The results obtained by many participants (in many small tasks) are aggregated into one global result, in our case, a ranking of 24 photographs, sorted in ascending order of how much money would be shared with the people represented on these photographs. As a result, we clearly see that physical appearance matters strongly, as it triggers known stereotypes such as 'poor' or 'rich', but potentially also many other stereotypes, most notably gender, but also attractiveness (facial expression). Race and age seem less important in this survey.

Although our results may sound rather obvious, eliciting and understanding these results would actually pose a major challenge to a computer, since implicit information is very hard to quantify. If we aim at quantifying implicit information, asking many people for their opinion is the only way of doing this without undesirable bias. Using COLLABORANK, we were able to quantify implicit information (i.e., physical appearance) to a degree, without introducing our own bias.

In future work, we would like to apply the methodology using a much larger set of photographs and a much larger set of participants. This may lead to further elicitation of various implicit factors that influence human decisions concerning fair shares. Such insights may also be valuable for designers of multi-agent systems that are meant to interact with humans; clearly, human strategies and preferences need to be understood as well as possible in this case.

References

[1] K. Basu. The Traveler's Dilemma. *Scientific American*, Volume 296, Number 6:68–73, 2007.

[2] S. Bowles, R. Boyd, E. Fehr, and H. Gintis. Homo reciprocans: A Research Initiative on the Origins, Dimensions, and Policy Implications of Reciprocal Fairness. *Advances in Complex Systems*, 4:1–30, 1997.

[3] G. Charness and M. Rabin. Understanding Social Preferences with Simple Tests. *Quarterly Journal of Economics*, 117:817–869, 2002.

[4] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodriguez-Aguilar, and P. Sousa. Issues in Multiagent Resource Allocation. *Informatica*, 30:3–31, 2006.

[5] W. Cohen, R. Schapire, and Y. Singer. Learning to order things. *Intelligence Research*, 10:243–270, 1999.

[6] A. Dannenberg, T. Riechmann, B. Sturm, and C. Vogt. Inequity Aversion and Individual Behavior in Public Good Games: An Experimental Investigation. *SSRN eLibrary*, 2007.

[7] S. de Jong, K. Tuyls, and K. Verbeeck. Fairness in multi-agent systems. *Knowledge Engineering Review*, 23(2):153–180, 2008.

[8] S. de Jong, K. Tuyls, K. Verbeeck, and N. Roos. Priority Awareness: Towards a Computational Model of Human Fairness for Multi-agent Systems. *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, 4865:117–128, 2008.

[9] P. G. Devine. Stereotypes and prejudice: Their automatic and controlled components. *Journal of Personality and Social Psychology*, 56:5–18, 1989.

[10] U. Endriss. Fair Division. Tutorial at the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2008.

[11] A. Falk and U. Fischbacher. A theory of reciprocity. *Games and Economic Behavior*, 54:293–315, 2006.

[12] E. Fehr. Don't lose your reputation. *Nature*, 432:499–500, 2004.

[13] E. Fehr and S. Gaechter. Fairness and Retaliation: The Economics of Reciprocity. *Journal of Economic Perspectives*, 14:159–181, 2000.

[14] E. Fehr and S. Gaechter. Altruistic punishment in humans. *Nature*, 415:137–140, 2002.

[15] E. Fehr and K. Schmidt. A Theory of Fairness, Competition and Cooperation. *Quarterly Journal of Economics*, 114:817–868, 1999.

[16] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, Princeton, USA, 2001.

[17] W. Gueth, R. Schmittberger, and B. Schwarze. An Experimental Analysis of Ultimatum Bargaining. *Journal of Economic Behavior and Organization*, 3 (4):367–388, 1982.

[18] C. Hauert, S. D. Monte, J. Hofbauer, and K. Sigmund. Volunteering as red queen mechanism for cooperation in public goods games. *Science*, 296:1129–1132, 2002.

[19] C. Hauert, A. Traulsen, H. Brandt, M. A. Nowak, and K. Sigmund. Via freedom to coercion: the emergence of costly punishment. *Science*, 316:1905–1907, 2007.

[20] J. Henrich, R. Boyd, S. Bowles, C. Camerer, E. Fehr, and H. Gintis. *Foundations of Human Sociality: Economic Experiments and Ethnographic Evidence from Fifteen Small-Scale Societies*. Oxford University Press, Oxford, UK, 2004.

[21] J. H. M. Janssens. Collaborative image ranking. Master's thesis, Faculty of Humanities and Sciences of the Universiteit Maastricht, 2008.

[22] G. Kalisch, J. W. Milnor, J. Nash, and E. D. Nering. Some experimental n-person games. Technical report, The Rand Corporation, U.S. Air Force, 1952.

[23] M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.

[24] D. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[25] M. Milinski, D. Semmann, and H. J. Krambeck. Reputation helps solve the tragedy of the commons. *Nature*, 415:424–426, 2002.

[26] H. Oosterbeek, R. Sloof, and G. van de Kuilen. Cultural Differences in Ultimatum Game Experiments: Evidence from a Meta-Analysis. *Experimental Economics*, 7:171–188, 2004.

[27] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 2 edition, 2003.

[28] F. C. Santos, J. M. Pacheco, and T. Lenaerts. Evolutionary Dynamics of Social Dilemmas in Structured Heterogeneous Populations. *Proc. Natl. Acad. Sci. USA*, 103:3490–3494, 2006.

[29] K. Sigmund, C. Hauert, and M. A. Nowak. Reward and punishment. *Proceedings of the National Academy of Sciences*, 98(19):10757–10762, 2001.

[30] L. von Ahn. Games with a Purpose. *IEEE Computer*, 39(6):92–94, 2006.

[31] S. C. Wheeler and R. E. Petty. The effects of stereotype activation on behavior: a review of possible mechanisms. *Psychol Bull.*, 127(6):797–826, 2001.

[32] T. Yamagishi. The provision of a sanctioning system as a public good. *J. Pers. and Soc. Psych.*, 51(1):110–116, 1986.

Discovering the Game in Auctions

Michael Kaisers [ab] Karl Tuyls [a] Frank Thuijsman [b] Simon Parsons [c]

[a] *Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven*
[b] *Maastricht University, P.O. Box 616, 6200 MD Maastricht*
[c] *Brooklyn College, City University of New York, Brooklyn NY 11210 USA*

Abstract

Auctions are pervasive in today's society. They provide a variety of markets, ranging from consumer-to-consumer online auctions to government-to-business auctions for telecommunications spectrum licenses. Starting from a set of trading strategies, this article enables a strategic choice by introducing the use of linear programming as a methodology to approximate heuristic payoff tables by normal form games. This method is evaluated on data from auction simulation by applying an evolutionary game theory analysis. The information loss in the normal form approximation is shown to be reasonably small such that the concise normal form representation can be leveraged in order to make strategic decisions in auctions. In particular, a mix of trading strategies that guarantees a certain profit against any population of traders is computed and further applications are indicated.

Keywords: Evolutionary game theory, Auction theory, Multi-agent games

1 Introduction

Auctions are deployed in a variety of real markets to foster highly efficient trading. They range from consumer-to-consumer markets like eBay via business-to-business stock exchanges to government-to-business auctions for mineral rights or government licenses for the telecommunications spectrum [7, 9]. Furthermore, auction mechanisms have been transfered successfully to solve other resource allocation problems, e.g. in the domain of efficient internet traffic routing [12]. Despite this diversity, even single markets of enormous scale may have profound impact on society, e.g. the New York Stock Exchange reported a trading volume of $11,060$ billion USD in 2000 [3]. This motivates researching how to run auctions and how to extract profit by trading within them as auctions are pervasive in today's society.

The traders that participate in an auction agree to subject to a set of market rules in order to exchange goods for money. Within the scope of this article only commodity markets are considered, i.e. a single type of an abstract good is traded. Each trader is assumed to have a *private valuation* of the good which is only known to himself. In double auctions, buyers and sellers place offers to indicate their intention to trade at a certain price [9]. The here considered *clearing house auction* proceeds in rounds and polls offers from each trader each round. When all offers are collected, an equilibrium price is established based on the available offers such that demand meets supply at this price. It is commonly set to the average of the two offers that define the range of possible equilibrium prices, i.e. the lowest bid and the highest ask that can be matched in the equilibrium. Each buyer with an offer above that price is matched with a seller having an offer below that price. The *profit* of a transaction can be computed as the difference between the transaction price and the private value, assuming that buyers will not buy above their private value and sellers will not sell below their private value.

A multitude of trading strategies has been devised to derive the next offer, possibly exploiting the knowledge about offers and transactions that were observed in previous rounds. The most trivial one is *Truth Telling* (TT) which just reveals the private value by placing offers exactly at that value. The experiment of this article instead considers three more sophisticated trading strategies which give raise to non-trivial evolutionary dynamics. Roth and Erev devised a reinforcement learning model of human trading behavior in [2] which is modified to perform in a clearing house auction as *Modified Roth-Erev* (MRE) [8]. MRE is

evaluated in competition to *Gjerstad and Dickhaut* (GD) and *Zero Intelligence Plus* (ZIP). GD maximizes the expected profit by computing the profit and probability of leading to a transaction for a set of relevant prices [5]. ZIP places stochastic bids within a certain profit margin, which is lowered when a more competitive offer was rejected and increased when a less competitive offer was accepted [1]. It can be noted that most trading strategies are adaptive to the progress of the auction, the internal processes however are outside the scope of this article.

Given a set of available trading strategies, it is of high interest which strategy is *best* in the sense that it yields the highest expected payoff. However, this question cannot be answered in general as the performance of a trading strategy is highly dependent on the competition it faces [13]. Let us therefore assume an auction that is populated by traders, each deploying one of the trading strategies above. The profit of each trader is dependent on the overall mix of strategies and traders may choose to change their strategy in the course of time. A *heuristic payoff table* is proposed in [14] and is adopted by several authors to capture the average profit of each type of trading strategy under all possible mixtures of strategies in a finite population [6, 10]. This table is a first step towards revealing the dynamics of adopted trading strategies in auctions and can for example be used to analyze which trading strategy yields the highest potential for improvements [10].

Although the heuristic payoff table provides the basis for analyzing the dynamics in auctions, it is unintuitive and lacks information about the payoffs for strategies that are not yet present in a population. However, exactly these payoffs would provide information about whether it is profitable or not to be the first one to adopt this strategy. The normal form game on the other hand enables an individual trader to calculate his expected profit for each of his possible choices against any mix of strategies he faces. It is more intuitive and allows inspecting the strategic situation with means from game theory, e.g. allowing to compute optimal strategies, best replies and Nash equilibria. This suggests the question whether a heuristic payoff table can be approximated by a normal form game in order to open up these opportunities. Answering this question affirmatively, this article demonstrates how an approximation can be found using linear programming, a common technique to optimize a linear goal function subject to a set of linear inequalities. The methodology is illustrated by approximating a heuristic payoff table from the auction domain and the results presented below show a reasonably small error such that the approximation can be leveraged for strategic considerations and an intuitive grasp of the game in auctions.

The remainder of this article is structured as follows: Section 2 introduces the game theoretical background that is required by the methodology presented in Section 3. Subsequently, Section 4 illustrates the method by applying it to an example from the auction domain and presents the resulting performance. These results are discussed in Section 5 and the paper is concluded with future directions in Section 6.

2 Game theoretical background

Classical game theory is the mathematical study of strategic conflicts of rational agents. Each individual i chooses a pure strategy s_i from the set of available strategies S_i and has a preference relation over all possible outcomes. The players are assumed to choose their actions simultaneously and independently. This implies that the preference relation can be captured by a numerical *payoff function* τ_i which is public knowledge and assigns a value of desirability to each possible joint strategy $s = (s_1, \ldots, s_n)$, where n is the total number of agents.

$$\tau_i : S_1 \times \ldots \times S_n \rightarrow \mathbb{R}$$

In the context of auctions, each pure strategy corresponds to a trading strategy and the preference relation is proportional to the profit that an agent can make given the set of opponents' trading strategies. This section introduces two different means to capture payoff functions for multi-agent games and auctions in particular: The normal form game and the heuristic payoff table. Furthermore, advantages and disadvantages of both representations are discussed. Subsequently, the evolutionary perspective is compared to the classical view from game theory and the concepts of replicator dynamics and basins of attraction are presented.

2.1 Normal form games

A normal form game commonly describes the payoff to each agent in matrix notation. The matrix given in Figure 1 describes a symmetric two-player normal form game. The first player may choose a row r, the second player chooses a column c and the joint choice (r, c) determines the payoff which the matrix gives for the first player. However, the payoff matrix for the second player equals the transposed of the first player's

	Rock	*Paper*	*Scissors*
Rock	0	−1	1
Paper	1	0	−1
Scissors	−1	1	0

Figure 1: Payoff matrix for the symmetric two-player normal form game Rock-Paper-Scissors.

payoffs in symmetric games. Hence, it can be derived from the same table by consulting the entry (c, r). Given k strategies, the matrix yields k^2 entries which compares favorably to the size of heuristic payoff tables given below. Both players seek to maximize their expected payoff; optimal strategies and the value of this game are derived in Section 5.2.

Within a normal form game, the player optimizes his expected payoff against an opponent that mixes according to a certain probability distribution. Similarly, he faces a field of traders that are distributed over the strategies in reality. It does not actually matter which opponent plays which strategy but rather how many opponents deploy which strategy. The opponent in normal form games therefore resembles the population in which the agent is situated in reality.

2.2 Heuristic payoff tables

A heuristic payoff table may also be used to capture the payoffs of a game [14]. However, it requires a finite population of traders such that all possible combinations of strategies can be evaluated. If each agent $i \in \{1, 2, \ldots, n\}$ has to choose a strategy $s_i \in \{1, 2, \ldots, k\}$, this leads to a joint strategy $(s_1, \ldots, s_n)$. However, for an individual trader it is only important to know how many of his opponents are playing each of the different strategies. So, given $(s_1, \ldots, s_n)$ the individual trader could derive that there are N_1 agents playing strategy 1, N_2 agents playing strategy 2, etc.. This would yield a *discrete profile* $N = (N_1, \ldots, N_k)$ telling exactly how many agents play each strategy. The average profit for playing a strategy can then be denoted by a payoff vector $U(N) = (U_1(N), \ldots, U_k(N))$ indicating that strategy $s \in \{1, 2, \ldots, k\}$ would yield an average payoff of $U_s(N)$ for the discrete profile N. The distribution of n agents on k pure strategies is a combination with repetition, hence the number of profiles in a heuristic payoff table is given by:

$$\binom{n+k-1}{n}$$

The payoffs of these discrete profiles can be measured in many domains, e.g. in auctions. However, measurements do not allow to capture the payoff to strategies that are not present, i.e. whenever $N_s = 0$ then $U_s(N)$ is unknown for that discrete profile. Table 1 shows the heuristic payoff table obtained from the experiments described in Section 4, indicating unknown payoffs with a dash.

The heuristic payoff table only delivers a snapshot of the game payoffs under fixed discrete profiles and does not yet assume adaptive agents. It is a merely descriptive representation of the conditions and the game dynamics need to be deduced using models on top of that. One opportunity to do so is the approximation of

Table 1: The first six rows give the transposed heuristic payoff table of a clearing house auction with 6 agents and the three strategies ZIP, MRE and GD. Each column gives a discrete profile N over the trading strategies and the corresponding payoff vector $U(N)$. Below, the deviation for the reconstructed payoff table from the normal form game representation is given which is discussed in Section 4. It features a maximal absolute deviation of 6.64% and a root mean squared error of 2.96%.

N_{ZIP}	6	5	5	4	4	4	3	3	3	3	2	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0
N_{MRE}	0	1	0	2	1	0	3	2	1	0	4	3	2	1	0	5	4	3	2	1	0	6	5	4	3	2	1	0
N_{GD}	0	0	1	0	1	2	0	1	2	3	0	1	2	3	4	0	1	2	3	4	5	0	1	2	3	4	5	6
U_{ZIP}	99	97	89	96	90	85	97	87	85	76	97	91	84	78	62	97	93	86	73	73	56	-	-	-	-	-	-	-
U_{MRE}	-	100	-	94	88	-	92	90	80	-	96	91	83	70	-	97	89	84	71	57	-	94	91	84	75	65	43	-
U_{GD}	-	-	69	-	65	69	-	64	73	73	-	66	67	76	80	-	62	69	75	77	80	-	62	67	71	76	79	79
ΔU_{ZIP}	6	3	3	1	3	6	0	−2	4	3	−2	1	2	4	−4	−3	2	2	−3	6	−3	-	-	-	-	-	-	-
ΔU_{MRE}	-	5	-	−1	3	-	−4	4	4	-	−1	4	7	3	-	−1	1	7	4	0	-	−4	2	6	6	6	−5	-
ΔU_{GD}	-	-	1	-	−2	−1	-	−2	3	0	-	1	−1	4	4	-	−2	1	4	3	1	-	−1	0	1	2	2	−2

the heuristic payoff table by a normal form game which allows to draw on the classical means from game theory to derive strategic choices (see Sections 3.2 and 5.2).

2.3 Replicator dynamics

Replicator dynamics describe game dynamics from an evolutionary perspective. Evolutionary game theory assumes an infinitely large population of individuals that choose their pure strategy according to some probability distribution. It assumes this population to evolve such that successful strategies with higher payoffs grow while less successful ones decay and it suggests to analyze the asymptotic behavior [4].

Evolutionary game theory takes a rather descriptive perspective replacing hyper-rationality from classical game theory by the concept of natural selection from biology. The evolutionary pressure by natural selection can be modeled by the replicator equations. Symmetric games only require the single-population replicator dynamics that define the growth of a strategy proportional to the fraction of the population that already uses this strategy and the difference between the payoff to this strategy and the average payoff.

The game dynamics for the normal form game with payoff matrix A can be calculated for strategy i given that the opponent mixes over the pure strategies according to the probability vector x:

$$\dot{x_i} = x_i \cdot [(Ax)_i - xAx] \tag{1}$$

It is also possible to construct the replicator dynamics from the heuristic payoff table immediately. To achieve this, the probability of each discrete profile for a finite population where each agent independently chooses its strategy according to some probability distribution needs to be calculated. This is a multinomial process for which the probability of the discrete profile N given the mixed strategy p can be computed as:

$$Pr(N|p) = \binom{n}{N_1, \dots N_k} \cdot p_1^{N_1} \cdot \ldots \cdot p_k^{N_k} \tag{2}$$

The payoff for each strategy can then be computed as the weighted average over the payoffs received in all profiles. However, a correction term is required if the payoffs for non-occurring strategies are unknown in the heuristic payoff table.

$$U_{average,i} = \frac{\sum_N Pr(N|p) \cdot U_i(N)}{1 - Pr(unknown|i)} \tag{3}$$

The resulting dynamics can be visualized in a vector field plot as in Figure 2 where the arrows indicate the direction of change and the length of an arrow is proportional to $|\dot{x}|$.

The replicator dynamics give rise to a dynamical system which may feature repellers and attractors of which the latter are of particular importance to the analysis of asymptotic behavior. Each attractor consumes a certain amount of the strategy space that eventually converges to it - this space is also called the basin of attraction. Assuming that an evolutionary process may start uniformly at any point in the strategy space, the size of the basin of attraction may be used to estimate the practical importance of an attractor. This can be achieved by uniform sampling of the strategy space and analysis of trajectory convergence, or under the assumption of continuous dynamics by grid sampling of the strategy space and a graph labeling algorithm. The latter applied to the given replicator dynamics leads to the basins of attraction depicted in Figure 4. A good approximation of the game dynamics should naturally have minimal impact on the basins of attraction.

3 Methodology

The conversion of a normal form game to a heuristic payoff table is straight forward. This gives raise to the question whether the inverse is also possible. However, the inverse transformation is over constrained and a heuristic payoff table can only be approximated by a normal form game. This section presents the newly proposed method to use linear programming for finding a suitable normal form game approximation.

3.1 From normal form games to heuristic payoff tables

The heuristic payoff table lists all possible discrete profiles with the average payoff of playing against a finite population that mixes accordingly. The payoff vector against the mixed strategy p can be computed from

the payoff matrix A as Ap. Given a matrix D where each row corresponds to a discrete profile N we can compute the matrix U that then yields the corresponding payoff vectors $U(N)$ as rows:

$$U = \frac{1}{n} \cdot D \cdot A^T \tag{4}$$

The heuristic payoff table is the compound of the discrete profiles D and the corresponding average payoffs given by U.

3.2 From heuristic payoff tables to normal form games

While the previous section has shown the transition from a normal form game to a heuristic payoff table, this section will reverse this step. However, the equation cannot simply be reversed as the values in the heuristic payoff table may be noise-prone due to stochasticity in the experiments and may also feature non-linear dynamics. Although this leads to an over-constrained system of equations, an approximation with minimal maximum absolute deviation can be found using linear programming.

Linear programming optimizes a linear goal function subject to a system of linear inequalities. The following program can be formulated in order to approximate a heuristic payoff table, where D is a matrix with all discrete profiles as rows, U is a matrix that yields the payoff vectors corresponding to D, $P = \frac{1}{n} \cdot D$ maps the discrete profiles to probabilities and M is the payoff matrix of the game that we search:

$$\begin{aligned} \text{min. } & \epsilon \\ \text{s.t. } & P \cdot M^T \leq U + \epsilon \\ & P \cdot M^T \geq U - \epsilon \end{aligned} \tag{5}$$

However, this notation needs to be transformed to standard notation in order to apply common algorithms from the linear programming literature. Let $c = (1, 0, 0, \ldots, 0)$, $x = (\epsilon, M_i)$, $A = \begin{pmatrix} -1 & \\ \vdots & P \\ -1 & \\ & -P \end{pmatrix}$ and $b = \begin{pmatrix} U_i \\ -U_i \end{pmatrix}$ where M_i is the i'th row of the payoff matrix to find and U_i is the i'th column of the payoff matrix. Then, this linear program can be solved in standard notation:

$$\begin{aligned} \text{min. } & c \cdot x^T \\ \text{s.t. } & A \cdot x^T \leq b \end{aligned} \tag{6}$$

In order to approximate the heuristic payoff table, we need to solve k linear programs to compute the complete normal form matrix.

4 Experiments

This section presents the experimental setup and results of measuring the information loss in the normal form game approximation of a heuristic payoff table from the auction domain. The heuristic payoff table given in Table 1 is obtained by simulating auctions with the *Java Auction Simulator API* (JASA) [11]. This empirical platform contains the trading strategies ZIP, MRE and GD which were set up with the following parameters, according to [1, 5, 8]: ZIP uses a learning rate of 0.3, a momentum of 0.05 and a JASA specific scaling of 0.2, MRE chooses between 40 discrete prices using a recency parameter of 0.1, an exploration of 0.2 and scaling of 9 and GD evaluates prices up to 360.

The heuristic payoff table is obtained from an average over 2000 iterations of clearing house auctions with 6 traders. On the start of each auction, all traders are initialized without knowledge of previous auctions and with a private value drawn from the same distribution as in [14], i.e. an integer lower bound b is drawn uniformly from $[61, 160]$ and the upper bound from $[b + 60, b + 209]$ for each buyer. The sellers' private values are initialized similarly. These private values then remain fixed over the course of the auction which runs 300 rounds on each of 5 trading days where each trader is entitled to trade one item per day.
The heuristic payoff table is approximated with a linear program as described in Section 3.2 which leads to the normal form game representation given in Figure 3. This normal form representation is transformed

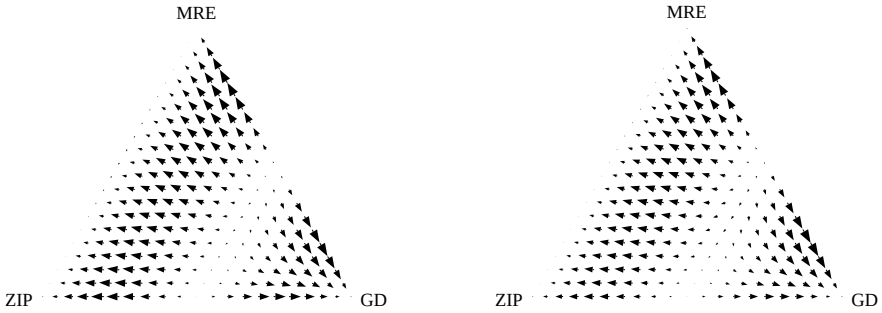


Figure 2: This figure shows the original replicator dynamics from the heuristic payoff table (left) and those from the normal form game approximation (right) in the clearing house auction.

	ZIP	MRE	GD
ZIP	93.8	102.7	52.9
MRE	94.9	100.0	38.3
GD	66.2	60.5	81.8

Figure 3: The symmetric two-player normal form game approximation of the heuristic payoff table for a clearing house auction with the three strategies ZIP, MRE and GD.

back into a heuristic payoff table as described in Section 3.1 and compared to the original table, leading to the differences indicated by ΔU_s in Table 1. Furthermore, the replicator dynamics are derived from both tables and compared in Figure 2. A more intuitive interpretation is given in Figure 4 which shows the basins of attraction that arise from the replicator dynamics.

Figure 2 shows that the differences in the replicator dynamics are very small and can hardly be observed by inspection of the vector field plots. Therefore, Figure 4 visualizes the basins of attraction which show a clear qualitative correspondence of the dynamics in the observed heuristic payoff table and the reconstruction from the normal form approximation. Only one mixed attractor moves slightly but this hardly impacts its basin of attraction. In the context of evolutionary game theory, evolutionary stable strategies provide a concept to find stable solutions in normal form games. The attractor $(0, 0, 1)$ which corresponds to a pure population of GD and the attractor $(0.7, 0.3, 0)$ are evolutionary stable in the normal form game and predict the attractors that are observed in the auction game dynamics.

5 Discussion

The results show that heuristic payoff tables in the domain of auctions may be approximated by normal form games with a reasonably small error. However, this case study is rather a proof of concept and does not necessarily generalize. Therefore, this section starts with a discussion of the limitations of this approach. Eventually, leveraging the newly gained insights for strategic choice is illustrated.

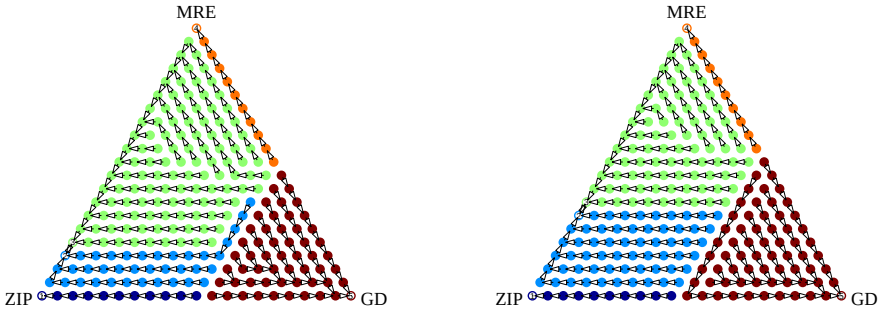


Figure 4: This figure shows the basins of attraction of the heuristic payoff table (left) and of the normal form game approximation (right) in the clearing house auction.

N_{ZIP}	5	4	4	3	3	3	2	2	2	2	1	1	1	1	1
N_{MRE}	0	1	0	2	1	0	3	2	1	0	4	3	2	1	0
N_{GD}	1	1	2	1	2	3	1	2	3	4	1	2	3	4	5
$U_{average}$	74.9	72.3	74.1	70.7	76.4	74.0	73.9	72.5	76.8	74.5	71.3	74.1	74.5	76.1	72.5

Table 2: The average payoffs of the optimal strategy $\pi^* = (0.3, 0, 0.7)$ against all discrete profiles for which the payoffs for the strategies ZIP and GD are known. The lower bound is given by 70.7 against the profile $N_{min} = (3, 2, 1)$ and the average is 73.9.

5.1 Limitations of the approximation

The proposed approach can be applied to any number of strategies. However, the approximation of heuristic payoff tables by normal form games imposes a less complex model on the data, which may be an oversimplification for the actual dynamics. Consequently, the precision of the approximation is likely to deteriorate when the number of trading strategies to choose from is increased.

5.2 Strategic choice

Consider the normal form representation of the auction game as given in Figure 3. It is possible to derive an *optimal strategy* that gives a lower bound on the profit that can be guaranteed even if nothing is known about the opponents. This profit is also known as the matrix game value and can be determined with standard algorithms from game theory. It equals 73.1 for this example and can be guaranteed by the optimal strategy $\pi^* = (0.3, 0, 0.7)$. This means that a rational trader who is playing ZIP with probability 0.3 and GD with probability 0.7 will get an expected payoff of at least 73.1 against any opponent that mixes between ZIP, MRE and GD. For any other probability distribution than π^*, he may encounter an opponent that gives him a lower expected payoff[1].

In order to validate these results, the optimal strategy may be applied to compute the average payoff against the distributions given in the heuristic payoff table. However, we need to restrict the consideration to those profiles for which all trading strategies that are used in the optimal strategy are present. Only then, the payoffs are known and the average can be computed accurately from the heuristic payoff table. These average profits are given in Table 2 for all these profiles, resulting in a minimum of 70.7 which is close to the approximated 73.1. Furthermore, the average overall profit against a uniformly distributed population is as high as 73.9. Calculation of the matrix game value and the optimal strategy can be achieved by linear programming since the agent simply wants to choose his probabilities in such a way that the minimal payoff over all columns is maximized, i.e. he is maximizing a linear function with respect to some linear inequalities.

If an agent knew the current actual mix of trading strategies in the population he faces, he could even make more than with the optimal strategy because the optimal strategy is based upon a worst case analysis. Under the assumption that all agents are have the same goal and the payoff matrix is known to all other agents as well, one can further assume that all other agents apply a similar reasoning and arrive at the same probability distribution. Therefore, a symmetric evolutionary equilibrium is more interesting in this context, especially if we want to understand how the distribution of traders will look like in the long run.

6 Conclusions

This article has modeled trading in auctions by considering a population of traders that repeatedly participate in an auction. A set of trading strategies is made available to the agents who make their choice according to the relative profit of these strategies.

The contributions can be summarized as follows: A methodology to approximate heuristic payoff tables by normal form games has been introduced. This smaller game representation is easier to analyze and fills in a gap of missing payoffs in the blind spots of the heuristic payoff table. Rather than merely participating myopically, a rational agent can now inspect the game strategically and reasoning from game theory can be applied.

[1] Note that the optimal strategy is not symmetric and therefore does not appear in the replicator dynamics.

The progress that has been made gives room for interesting opportunities. This approach needs to be tested on other auctions and domains, possibly applying it to higher dimensions as it is general in the number of strategies. This would allow inspecting the game in auctions that yield more than 3 strategies which cannot be visualized easily. Another direction extends the game theoretic analysis, e.g. investigate symmetric equilibria analytically in the normal form game. Furthermore, we aim to argue for the described approach on a more theoretical level and look for structure in the deviation from the linear model, in particular where and why qualitative changes occur.

7 Acknowledgments

The authors wish to express their gratitude to Jinzhong Niuand Steve Phelps for their support and helpful discussions that have contributed to the findings of this paper. This research was partially sponsored by a TopTalent 2008 grant of the Netherlands Organisation for Scientific Research (NWO).

References

[1] Dave Cliff and Janet Bruten. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical report, Hewlett-Packard Research Laboratories, 1997.

[2] Ido Erev and Alvin E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *The American Economic Review*, 88(4):848–881, 1998.

[3] New York Stock Exchange. Stock market activity. Technical report, New York Stock Exchange, 2000. available at http://www.nyse.com/ . . . pdfs/02_STOCKMARKETACTIVITY.pdf.

[4] H. Gintis. *Game Theory Evolving*. Princeton University Press, 2000.

[5] Steven Gjerstad and John Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29, January 1998.

[6] Tomas B. Klos and Gerrit Jan van Ahee. Evolutionary dynamics for designing multi-period auctions (short paper). In *Proceedings 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*. IFAAMAS, 2008.

[7] John McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8(3):145–162, Summer 1994.

[8] J. Nicolaisen, V. Petrov, and L. Tesfatsion. Market power and effciency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523, 2001.

[9] Simon Parsons, Juan Rodriguez-Aguilar, and Mark Klein. A bluffer's guide to auctions. Technical report, Center for Coordination Science, MIT, 2004.

[10] S. Phelps, M. Marcinkiewicz, and S. Parsons. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 705–712, Hakodate, Japan, 2006. ACM.

[11] Steve Phelps. Java auction simulator api. http://www.csc.liv.ac.uk/ sphelps/jasa/, 2005.

[12] Tim Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[13] J. Rust, J. Miller, and R. Palmer. Behavior of trading automata in a computerized double auction market. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley, 1993.

[14] William E. Walsh, Rajarshi Das, Gerald Tesauro, and Jeffrey O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In Piotr Gmytrasiewicz and Simon Parsons, editors, *Proceedings of the Workshop on Game Theoretic and Decision Theoretic Agents*, pages 109–118, 2002.

Maximizing Classifier Yield for a given Accuracy

Wessel Kraaij [a] Stephan Raaijmakers [a] Paul Elzinga [b]

[a] *TNO Information and Communication Technology*
Delft, The Netherlands
[b] *Regional Policeforce Amsterdam Amstelland*
Amsterdam, The Netherlands

Abstract

We propose a novel and intuitive way to quantify the utility of a classifier in cases where automatic classification is deployed as partial replacement of human effort, but accuracy requirements exceed the capabilities of the classifier at hand. In our approach, a binary classifier is combined with a meta-classifier mapping all decisions of the first classifier that do not meet a pre-specified confidence level to a third category: *for manual inspection*. This ternary classifier can now be evaluated in terms of its *yield*, where yield is defined as the proportion of observations that can be classified automatically with a pre-specified minimum accuracy.

1 Introduction

The evaluation practice of information processing tasks such as classification, detection and ranking is a non-trivial issue, where no ideal recipe exists. Evaluation is either tailored toward component benchmarking or can be focused on end-to-end user experience. The component evaluations have their roots in the Cranfield Information Retrieval experiments that were a model for the successful TREC evaluations [10]. These batch style experiments have for a long time focused on automatic only experiments, where human involvement is separated as much as possible from the actual experiments in order to avoid inter user variability and completely focus on the actual system component under scrutiny. Such batch style experiments have been attractive for IR researchers and even inspired evaluations in other communities such as natural language processing, since experiments were easy to conduct, and also very economic because humans were excluded from the loop (except for creating the ground truth). Still many researchers felt that these studies were limited, since they failed to model a real search process.

The component based evaluation which is the model for TREC is sometimes referred to as intrinsic evaluation in contrast to an evaluation where the component's performance is measured in the user context (extrinsic). When evaluating a complete system, intrinsic evaluation approximates *performance* evaluation and extrinsic evaluation is related to *adequacy* measurement [4]. In such a task based evaluation, factors such as usability play a crucial role. Performance measurements are usually aimed at comparing systems, whereas adequacy measurements focus more on the usability and practical use for an end user.

In many scenarios, the classification accuracy of a machine learning based classification system is not sufficiently high, since the tasks at hand are difficult. We propose that for these scenarios, systems can still successfully be deployed if only the "easy cases" are classified automatically. In such a deployment scenario, quality standards can still be met, whilst reducing (and not completely replacing) the manual workload.

The objectives of this paper are two-fold:

1. Introduce a novel ensemble of classifier evaluation measures which can evaluate the deployment of a classifier which only partially replaces human labeling.

2. Develop a ternary classifier that can operate at a pre-specified accuracy by forwarding "difficult" items for manual processing.

	assigned class: "+"	assigned class: "–"
ground truth: "+"	TP	FN
ground truth: "–"	FP	TN

Table 1: Classification contingency table. Precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$.

In this paper we propose a novel ensemble of evaluation measures for classification tasks that can be used for component evaluations. The distinguishing characteristic of this new ensemble is the fact that both measures (accuracy and yield) are motivated from the task viewpoint and directly relate to potential cost savings in terms of reduced manpower. The structure of this paper is as follows: in section 2 we give a formal definition of the new ensemble of evaluation measures and discuss the relationship of these measures with operational characteristics of an abstracted workflow (an office where analysts manually label documents). Section 3 illustrates the ensemble of measures by reporting experiments concerning automatic detection of domestic violence cases in police files and a spam detection task. Section 4 describes the ternary classifier architecture. Section 5 presents two experiments that illustrate the value of the evaluation method and the ternary classifier. The paper concludes with a discussion section.

2 Classifier accuracy and classifier yield

Several evaluation measures dominate the field of component based evaluation for classification and ranking tasks. The field of information retrieval evaluation popularized the precision and recall measures. These are set based measures which can best be visualized by looking at a contingency table (Table 2). Whereas the original precision and recall measures are hardly used anymore in IR (instead mean average uninterpolated precision is the norm for ranking tasks), they are regularly reported for classification experiments. Precision and recall have the desirable property that they relate well to intuitive characteristics of quality. Better systems have higher precision and or recall values. A disadvantage of precision and recall is that the test set must be a representative sample of the real class population. An opposite approach is to quantify the error rates of a classifier, where a better system has smaller error rates. For a binary classifier scenario both type I and type II error rates (false alarms and misses) can be measured independently from the actual class distribution in the test set.

Precision is a measure of fidelity and is inversely related to type I errors (false positives). Recall can be seen as a measure of completeness, being inversely related to type II errors (false negatives). An important nuance to make here is that fidelity and completeness are defined with respect to the positive class label, i.e. the task modeled is correctly identifying items with a positive class label. Precision and recall can be combined into a single measure F_β [9], which helps to compare systems at a certain operating point (usually precision and recall are considered equally important). Note that precision and recall are defined from the perspective of the positive class.

Another measure that is often reported for classifier evaluation experiments is classifier accuracy. This is an intuitive measure for classification quality provided the class prior probabilities do not differ too much. The accuracy quantifies the quality of both positive and negative decisions made by the (binary) classifier. This averaging behaviour makes accuracy highly sensitive to a skewed distribution of class priors (imbalanced natural class distribution). This means that it is difficult to interpret accuracy results unless the class distribution of the test set is known. A simple majority classifier can have a very high accuracy for skewed distributions.

A subclass of typical real-life classification problems are detection tasks. These can be characterized as binary classification tasks with a skewed natural class distribution i.e. the negative cases are much more common than the positive cases. We are aware of the problems that these kinds of tasks pose for training classifiers and for designing benchmark data sets (some of these issues were briefly introduced above). A training data set needs to contain sufficient positive examples of a relatively rare phenomenon. The test data set however should contain enough negative examples in order to have a proper estimate of false positives. These are all important issues for the design of evaluations, but they are not the focus of this paper. Our claim is that just stating that a classifier has a certain F_1 value or accuracy cannot be translated in terms of its potential for operational deployment. Also, in some scenarios the problem is so difficult that state of

	assigned class: "+"	assigned class: "–"	assigned class: "?"
ground truth: "+"	TP	FN	M
ground truth: "–"	FP	TN	

Table 2: Classification contingency table for the ternary classifier

the art classifiers do not meet the minimum quality requirements that have been defined for this task. Still, if we could modify the workflow of human analysts and the classifier architecture in such a way that part of their work could be automated, while meeting the minimum quality requirements, it is easy to define a business case. We therefore propose a novel and intuitive way to quantify the utility of a classifier in cases where classification is applied in order to partially replace human labour, but accuracy requirements exceed the capabilities of the classifier at hand. Typical application scenarios are binary detectors. In our approach, a binary classifier is combined with a meta-classifier mapping all decisions of the first classifier that do not meet a pre-specified confidence value to a third category: *for manual inspection*. The classifier combination can be seen as a ternary classifier, which can now be evaluated in terms of its *yield* at a pre-specified confidence level, where yield is defined as the proportion of observations that can be classified automatically with a minimum pre-specified accuracy. In a way, accuracy and yield model the same intuitive aspects that underly precision and recall, classifier accuracy is a way to measure the fidelity of the classification task and classifier yield can be viewed as a measure for classifier completeness at the task level. The intended use of the ensemble $\{$*accuracy,yield*$\}$ is to measure the classifier yield at a fixed (minimum) level of accuracy. As an example, we could be interested in the yield of a biometric detector at an accuracy level of 99%.

Table 2 shows a modified contingency table where the classifier can assign one additional label: "?" (queue for manual inspection). Now accuracy can be defined as usual:

$$\textit{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

and yield can be defined as:

$$\textit{yield} = \frac{TP + TN + FP + FN}{TP + TN + FP + FN + M} \tag{2}$$

It is easy to see that the classifier yield is just the proportion of observations that is not labeled as M.

3 Related work

As far as we know, the proposed ensemble of measures (yield at minimum accuracy) is a novel way of measuring the quality of a classifier. There are several established evaluation traditions that have some elements in common. The TREC filtering task used a linear utility function for the adaptive filtering task, which is a rather complex classification task where a system can use feedback in order to set its optimal operating point (decision threshold) in a dynamic fashion. The linear utility is defined as [5]:

$$\textit{linear utility} = \alpha \times TP + \beta \times FP + \gamma \times FN + \delta \times TN \tag{3}$$

This is essentially a cost function, where parameters must be chosen to model a particular user scenario. Choosing four parameters (which can be negative) is non-trivial, and therefore in our view not so intuitive. Linear utility could be extended to handle the five-cell contingency table corresponding to our ternary classifier, but that would mean five parameters to choose. A more elegant way to model the *cost* of running a certain classifier on a dataset is the family of cost functions that were developed in the Topic Detection and Tracking (TDT) framework [3]. The basic cost function is defined as follows:

$$\textit{detection cost} = C_{Miss} \times P_{Miss} \times P_T + C_{FA} P_{NT} P_{FA} \tag{4}$$

where C_{Miss} and C_{FA} are fixed cost parameters that tax type II and type I errors respectively, P_{Miss} and P_{FA} are the probabilities (normalized counts) of type II and type I errors (false alarms), and $P_T = 1 - P_{NT}$ is the prior probability of a positive class label (*T=target*). Usually, the detection cost is measured at different levels of Miss/False Alarm trade-off by threshold sweeping, thus generating a detection cost curve. The detection cost function is motivated by the desire to quantify different types of error and sum the complete cost of a detection task for a certain data collection (taking into account the relative proportion of the class

population sizes). However, the detection cost is based on a fully automatic scenario. Incorporating the cost of manually assessing observations would make the detection cost function less intuitive.

Another common aggregate statistic for measuring classification is the AUC (area under (ROC) curve) [2]. AUC is the ROC (receiver operating curve) equivalent of mean average uninterpolated precision. ROC is based on a plot of the true positive rate (recall) versus the false positive rate. ROC curves are less optimal for unbalanced classes, since the interesting part of the curve needs zooming [6]. In principle it should be possible to use our ternary classifier architecture for a yield fixed AUC evaluation scenario, although AUC is not a very intuitive quality measure for non-experts.

Finally, a common evaluation procedure for biometric detectors is to measure the false alarm rate (FAR) at a fixed maximum false reject (miss) rate (FRR) or vice versa [1]. Our proposed procedure is similar in the respect that a certain operating point is pre-defined in order to compare systems. The pre-defined operating point provides an "anchor" in the recall-precision trade-off and simplifies evaluation to a single measure just like F_β defines a certain operating point in the precision recall space.

4 An example ternary classifier

The experiments that were carried out to illustrate the evaluation procedure were based on a two-level classifier architecture. The first level classifier was implemented by an information diffusion kernel machine. This kernel machine presupposes L1-normalized data (relative frequencies) and estimates similarity between documents using a geodesic distance measure applied to the Riemannian manifold that represents this data [11]. The (parameter free) diffusion kernel machine was modified to provide a posterior probability as output in addition to the predicted class [7]. The mapping function was trained on a separate development data set. The posterior probability (Platt score) was subsequently used as an input score σ for a meta-classifier that was implemented by a decision rule based on two thresholds θ_l and θ_u. The decision rule was defined as follows:

$$\mathit{prediction}(\sigma) = \begin{cases} + & \mathit{if} \quad \sigma => \theta_u \\ M & \mathit{if} \quad \theta_l < \sigma < \theta_u \\ - & \mathit{if} \quad \sigma <= \theta_l \end{cases} \tag{5}$$

The thresholds maximizing the yield while satisfying the pre-specified minimum accuracy were computed through exhaustive search by a two dimensional parameter sweep (for both threshold parameters θ_u and θ_l) on a development set.

The development data set for parameter training should be chosen carefully since we assume that the class distribution is the same in the development set and the test set and that the Platt score distribution is more or less similar in the development and test set, for both classes.

5 Experiments

We will illustrate the use of the evaluation procedure by two experiments. The first experiment concerns the detection of domestic violence in police files. The second experiment is about spam detection

5.1 Detection of domestic violence

Taking adequate action in cased of domestic violence is one of the focal points of the regional police force Amsterdam-Amstelland (RPAA). Recognition of domestic violence as such in incident reports is not an easy task, since domestic violence has a complex legal definition where several conditions need to be checked. Domestic violence is not always marked as such in the reports by the registrating police officer, so it is desirable to recognize these cases post-hoc automatically. The current practice for filtering out domestic violence cases from the full database of incident reports is based on a rule based system. Rules are created and maintained manually. Unfortunately the current rule set creates a very high number of false positives, which means that all filtered cases currently are subjected to a manual check. In order to minimize the number of manual checks, two classifiers were compared on site. A baseline rule based classifier[1] using hand-crafted thesauri (more elaborate and refined than the incident-report filtering system) and the ternary

[1] This classifier is actually a ranking system, where a decision threshold was chosen manually.

	accuracy	yield
baseline classifier	0.73	1
diffusion kernel machine	0.84	1

Table 3: Results for the detection of domestic violence on the full test set using a single classifier

	accuracy	yield
development set	0.90	0.70
full test set	0.92	0.86
test set sample A	0.93	0.86
test set sample B	0.92	0.89
test set sample C	0.93	0.86

Table 4: Results for the detection of domestic violence experiment using the ternary classifier

classifier discussed in Section 4. The ternary classifier architecture used the same feature set as the baseline classifier. Example features are *my father beats* and *my uncle abducts*, where verb forms were normalized.

The evaluation procedure based on accuracy and yield was applied in order to provide simple intuitive statistics that would enable a transparent interpretation of what a deployment of an automatic classifier would mean in terms of reduction of processing time, whilst maintaining the required quality level.

The following datasets were used:

training set A collection of 1736 reports, manually re-checked. 1101 positive cases. A random sample of 200 case files was used for development, the rest (1536) for training.

test set A held out collection of 2291 reports, labeled by registrating officer. 541 positive cases

As a first step the diffusion kernel and Platt function were trained on the development set. In a second step, optimal upper and lower decision score threshold were computed using the development data with a pre-specified accuracy > 0.90.

Table 3 lists the evaluation results (measured in terms of accuracy) for the baseline rule based ranking classifier and the diffusion kernel machine. The more advanced classifier architecture has a superior performance thanks to its generalizing capabilities. Still the accuracy of the diffusion kernel machine is too low for deployment at RPAA. In a second step, score thresholds are learned on a development set[2] to isolate those reports where the classifier decision is based on a low confidence score. These reports can then be forwarded for manual inspection. As an illustration, Figure 1 shows the probability that the classifier is correct as a function of its score.

The important question is whether decision thresholds can be learned and whether they are robust. Table 4 lists the accuracy and yield of the ternary classifier for development and test sets. As an additional diagnostic, three random samples of the test set (sample size = 1000) were evaluated. The obtained accuracy and yield on the test set are both higher than on the development. This could be explained by the fact that the test set was obtained from cases from a different year, where annotation standards might have changed. Still, results of the classifier on development and test set show the potential of the proposed approach, which seeks to minimize the amount of human labeling while meeting pre-specified quality standards. The results at various subsamples demonstrate the robustness of the parameter settings. Related work on the same dataset explores the possibility of involving a human expert for an interactive selection and definition of complex features, based on formal concept analysis [8].

5.2 Spam detection

As a second experiment we chose a spam detection task, available from the ECML 2006 Discover Challenge `http://www.ecmlpkdd2006.org/challenge.html`. The challenge consists of two separate tasks: a task (A) with many user-specific training data addressing user-specificity of the found solution, and a task (B) with a limited amount of data per user, addressing generalization over users. In this work, we

[2]We did some preliminary experiments varying the size of the development set and a size of 100 was still sufficient.

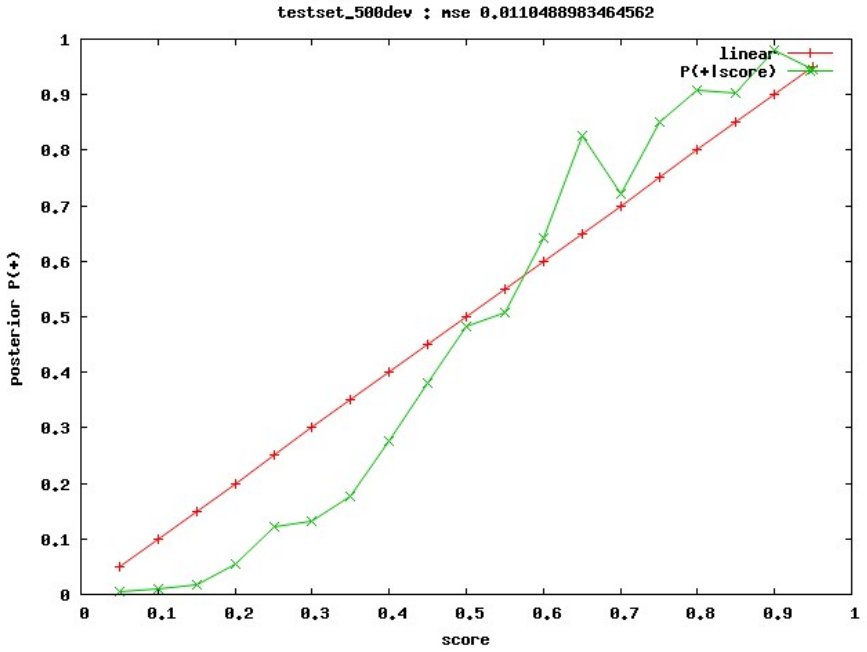


Figure 1: Posterior probability as a function of Platt score

	#pos dev	#pos test	binary classifier accuracy	ternary classifier accuracy	ternary classifier yield
user 0	248	1002	0.62	0.89	0.19
user 1	241	1009	0.65	0.90	0.39
user 2	268	982	0.78	0.91	0.69

Table 5: Results for the detection of spam emails using a binary and ternary classifier

limit ourselves to task A. All data sets consist of word/frequency pairs, which can be easily normalized to L1.

Task A models three users. For each user there are 4000 labeled training email messages and 2500 for evaluation. We divided the evaluation sets in a development set of 500 emails and the remaining 2000 for evaluation.

Table 5 lists the results of the spam detection experiment. The first two columns give the number of spam messages in the development and test set respectively. The third column gives the accuracy of the standard binary classifier (diffusion kernel machine). The fourth and fifth column give results on accuracy and yield when the ternary classifier's thresholds have been set for a minimum accuracy level of 0.90 using the development subsets. The desired accuracy (0.9) can be achieved for about 20-70% of the email messages depending on the user, making it a much harder task than the domestic violence detection.

Figure 2 illustrates the optimal operation curves for each user mailbox in a so-called *yieldplot*, where the classifier yield is plotted as a function of the desired accuracy level.

6 Discussion and Conclusions

We have presented a new ensemble of evaluation measures for a setting where a classifier is used to partially replace human labelling effort. The measures accuracy and yield relate well to a more extrinsic view on evaluation, where the focus is on cost savings. Accuracy and yield can be seen as workflow oriented measures for 'fidelity' and 'completeness'. The simplicity of this approach does have some shortcomings. Indeed accuracy as an aggregated measure hides the different sources of classification quality. it is well known that accuracy is sensitive to class imbalance. An alternative ensemble based on false alarm rate, false reject rate

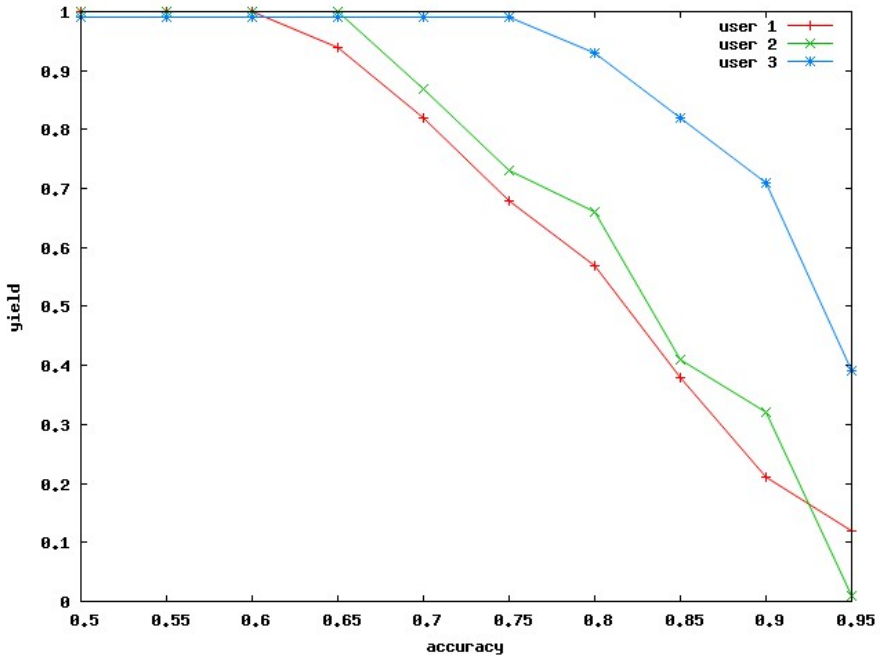


Figure 2: Yield as a function of (minimum) classifier accuracy, in the ternary classifier setting

and yield would solve this problem. However, this ensemble might be less intuitive for non-experts.

A second contribution of this paper is the concept of a ternary classifier, which forwards cases that cannot be classified with a pre-specified confidence to a human expert, thereby reducing the error rate of the classifier. Our method estimated two posterior probability threshold levels. The experiments show that the yield vs. accuracy plot makes it easy to use the ternary classifier in an operational workflow. Also, the ternary classifier can effectively forward difficult cases for human inspection.

In fact it is not essential that the classifier outputs true probabilities, it can be any monotonous increasing ranking function. As long as ranking values can be compared across collections, since the threshold values will always be optimized on a different data set than the test set.

There are several ways in which we plan to extend this research. We intend to look at the suitability of other (first level) classifier architectures, look at an ensemble of measures that makes a distinction between type I and type II error rates, and perform a more thorough analysis of the robustness of our parameter setting procedure.

References

[1] Ruud Bolle, Jonathan Connell, Sharanthchandra Pankanti, Nalini Ratha, and Andrew Senior. *Guide to Biometrics*. SpringerVerlag, 2003.

[2] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861874, 2006.

[3] Jonathan G. Fiscus and George R. Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking: event-based information organization*, pages 1731. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[4] L. Hirschman and H. S. Thompson. *Survey of the State of the Art in Human Language Technology*, chapter 13.1 Overview of Evaluation in Speech and Natural Language Processing. 1996.

[5] David A. Hull and Stephen E. Robertson. The TREC-8 filtering track final report. In *Proceedings of TREC-8*, 1999.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[7] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A.J. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classiers*, pages 6174, 2000.

[8] Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. An exploration into the power of formal concept analysis for domestic violence analysis. In Petra Perner, editor, ICDM, volume 5077 of *Lecture Notes in Computer Science*, pages 404416. Springer, 2008.

[9] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[10] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[11] Dell Zhang, Xi Chen, and Wee Sun Lee. Text classification with kernels on the multinomial manifold. In *SIGIR 05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266273, New York, NY, USA, 2005. ACM.

Stigmergic Landmarks Lead The Way

Nyree Lemmens [a] Karl Tuyls [b]

[a] *Faculty of Humanities and Sciences, MICC, Maastricht University, The Netherlands*
[b] *Faculty of Industrial Design, Eindhoven University of Technology, The Netherlands*

Abstract

In this paper, we describe a nature-inspired optimization algorithm based on bee foraging behavior. This algorithm takes the high performance of bee path-integration navigation and adds ant-like stigmergic behavior in the form of landmarks. More precisely, each individual landmark can be created at any unblocked state in the environment and contains a collection of direction markers with which visiting agents can find their way in an unknown environment. A landmark can either be represented by an agent or any other information-distributing object. Essentially, we implement ant recruitment behavior based on pheromone. However, instead of using attracting or repelling pheromone in every state of the environment, we only update directional information at key locations in the environment. The resulting algorithm proves to be very efficient.

1 Introduction

In previous work we presented Bee System as an alternative to Ant Colony Optimization in foraging domains [17]. Foraging is an interesting problem in the domain of Multi-Agent Systems (MAS). The problem entails that a group of agents has to gather objects (food) in a complicated, potentially dynamical environment. Ants deposit pheromone on the path they take during travel. Using this trail, they are able to navigate towards their nest or food. Ants employ an indirect recruitment strategy by accumulating pheromone trails. When a trail is strong enough, other ants are attracted to it (i.e., recruitment) and will follow this trail towards a destination. This is known as an autocatalitic process. More precisely, the more ants follow a trail, the more that trail becomes attractive for being followed. Short paths will eventually be preferred. Since ants follow pheromone trails towards their destination, it may be clear that the ant's navigation strategy is also guided by pheromone.

In contrast, non-pheromone-based algorithms are inspired by the behavior of (mainly) bees and do not use pheromones to navigate through unfamiliar worlds. Instead, for navigation, they use a strategy named Path Integration (PI). Bees are able to compute their present location from their past trajectory continuously and, as a consequence, can return to their starting point by choosing the direct route rather than retracing their outbound trajectory [20, 15]. For recruitment, bees employ a direct strategy by dancing in the nest. Their dance communicates distance and direction towards a destination [26].

In previous research, we introduced a foraging algorithm based on bees and compared it to an Ant System [17]. The comparison showed that the bee-inspired, non-pheromone-based algorithm clearly outperformed the ant-inspired, pheromone-based algorithm in relatively unobstructed environments; more precisely, the bee algorithm was able to solve the foraging problem about three times as fast as the ant algorithm [17]. However, we also found that in more constrained and/or dynamical environments, such as the Deneubourg Bridge [10], the bee algorithm tends to get stuck behind obstacles [17]. Unlike the ant algorithm, the bee algorithm is only able to learn from mistakes at one location (i.e., the hive) and as such is less adaptive.

To account for the adaptability problem we introduced inhibition pheromones, resulting in the algorithm 'Bee System with inhibition Pheromones' (BSP) [16]. More precisely, agents can place pheromones in cells that can be considered undesirable, for instance cells that lead to a dead end. In other words, this algorithm is a hybrid, integrating some features of Ant Systems into a bee-inspired algorithm. The major advantage of this algorithm is that its performance is equal to our initial Bee System algorithm while being at least as adaptive as the ant-inspired algorithm. However, the major downside of this approach is that the inhibition pheromones needs to be simulated somehow, which is not a trivial issue, especially in embodied systems.

In this paper therefore, we explore the possibility of using landmarks instead. With landmark navigation, agents can learn landmarks that together function as routes. Each landmark represents a segment in the total route [6] and indicates what follow-up action to take to get to the next segment. Since landmarks are naturally present in an environment, detection of such landmarks is inherently dynamic and obviously so is the adaptability of the agents. Moreover, due to route segmentation, landmark navigation ensures robustness and reduces accumulating navigation errors. Although landmark navigation is not a new research field, our research focuses on using landmark navigation in a swarming MAS which is novel research. More precisely, we extend our previously introduced Bee System [17] with landmark navigation. In this new algorithm, each agent is able to contribute to the success of a landmark route. The more agents follow a route, the more that route becomes attractive for being followed. Summarizing, we present a stigmergic landmark algorithm.

The remainder of this paper is structured as follows. First, we present an overview of related work in the area of landmark navigation. Next, we present our new landmark algorithm and show its effectiveness in a set of experiments. Finally, we conclude and look at future research.

2 Related Work

In this paper, we focus on landmark navigation in large groups. More precisely, we focus on swarm communities such as bee colonies. Individuals of such colonies are limited in their means to navigate over typically unknown environments due to limitations in, for example, brain capacity. However, where the individual fails, the swarm as a whole is able to perform complex tasks such as finding the shortest route to a food item.

In bee communities, landmark navigation is used as an additional navigation strategy to Path Integration [8]. It actually serves as a strategy to make PI navigation more accurate [2, 9]. Each member of the colony is able to segment the traveled path and measure its traveled distance by recognizing landmarks and acting on them once they are recognized. The distance that spans between two consecutive landmarks is represented by a so-called local vector. Such a vector has a direction and a magnitude component in which magnitude represents distance. Each local vector is attached to the landmark that is seen at the beginning of the segment, such that when a landmark is recognized, the bee recalls the associated local vector that guides it to the next landmark. Local vector memories can be linked together in a sequence so that, even in the absence of the triggering landmark, memory recall occurs [18, 5, 7, 6]. Each bee furthermore keeps an up-to-date global vector which spans between the start location and the destination [5]. However, when bees move over familiar territory, local vectors activated in a sequential fashion are preferred over the global path integration vector. The global vector thus serves as a backup strategy [4].

Bees basically classify landmarks into two groups, (i) global landmarks and (ii) local landmarks. First, global landmarks are used to guide the bee to the rough area of the goal and can be used over large spatial distances. This panorama (i.e., the global landmarks) determines which actions of movement are recalled [7, 25]. Second, local landmarks are close to the goal (or the hive) and are therefore a natural choice for pinpointing the goals position [3, 7, 25]. However, local landmarks are not as reliable as global ones. Local landmarks (e.g., a leaf) tend to change over time whereas global landmarks (e.g., a mountain range) tend to be stable. Research has shown that in absence of known global landmarks, bees are still able to navigate towards the goal albeit not very accurately. Local landmarks seem to be used as a secondary mechanism of navigation. The local landmarks are not attended when a familiar route and known global landmarks are present. Thus, bees preferably attend to global landmarks [25].

In recent years, researchers have started to use landmark navigation behavior to create robots which autonomously can navigate in unknown environments. In [1], a single robot is presented which builds an internal environment map in order to navigate through it. Although the robot's control system is designed as a MAS, it is solely used to determine which control system (for example, the pilot system) of the robot gets access to the robot. By using a bidding mechanism the most urgent control system gets access. In [13, 12], a single robot is able to navigate through an environment based on image matching. The robot is allowed to take snapshots while recording positional information. By mathematically comparing two snapshots, the robot is able to calculate an average displacement vector with which it can move towards a goal. Together these vectors make a displacement field with which a robot can move from any point in the environment towards its destination. [19] also demonstrates an image-matching approach to landmark navigation and shows its functionality in a robot with analog electronic hardware. [14] presents a single robot which uses the Perception-Action architecture which is a neural computation architecture proposed to solve a wide variety of control problems requiring learning capabilities. By using this architecture, the robot is able to learn how to reach any goal from any other landmark by moving around the goal. [21] presents a robot with a self-organizing mechanism to build a representation of the environment based on landmark recognition. The

robot first starts with a map-building effort and once completed the map interpretation starts which consists of map planning and execution. Note that all these researches are based on single robots. Furthermore, most researches are based on offline learning, i.e., a map has to be learnt before it can be used.

3 Algorithm

In this section we will describe the algorithm developed. First, for the sake of clarity, we give a short overview of the original bee-inspired algorithm from [17]. Next, we describe the novel MAS landmark navigation algorithm. In the following, when we speak of 'the hive' we indicate the starting position of the agent. 'The goal' indicates a destination with a collection of items present.

3.1 The original bee-inspired algorithm

The bee algorithm implements both recruitment and navigation behavior. *Recruitment behavior* is implemented in analogy with biological bees' dance behavior [22, 16, 17]. Agents share information on previous search experience (i.e., the direction and distance toward a certain goal) only when they are in the hive. Agents in the hive can then decide whether to exploit previous search experience obtained by other agents in the hive, or to exploit their own search experience, if available. Biological bees use a (still) unknown decision mechanism to decide whether to exploit another bee's experience. In our bee-inspired algorithm, the decision is based on distance assessment; an agent will exploit another agent's experience if it indicates goals at a shorter distance from the hive than the goal currently known by the agent. The *navigation behavior* either exploits previous search experience (of the agent itself or of another agent in the hive) or lets the agents explore the world using an exploration strategy called a Lévy flight [24]. Exploiting previous search experience is guided by a PI vector that agents either have constructed themselves or have adopted from another agent in the hive. In the remainder of this paper we refer to a goal-directed PI vector as a Goal Vector (GV), a hive-directed PI vector is referred to as a Homing Vector (HV).

The general structure of our bee-inspired algorithm is quite similar to that of algorithms in ACO [11]. It implements both recruitment and navigation behavior and consists of three functions. First, $ManageBeesActivity()$ handles agents' activity based on their internal state. Each agent is in one of six internal states. In each state a specific behavior is performed. (i) agent state 'Explorer' indicates the agent is exploring the environment for goals and performing Lévy flights to optimally cover the search space [23]. During exploration, the agent continuously updates its HV. (ii) agent state 'Carrier' indicates the agent found a goal and is returning an item from the goal to the hive. The actions an agent takes during the return are guided by the HV. During the return, the agent continuously updates its HV in order to have an accurate vector of guidance. The GV is set to the $HV + 180^\circ$ the moment the agent arrives at the goal. (ii) 'Dancer' indicates that an agent arrived at the hive with directional knowledge of a goal (i.e., the GV). In order to recruit other colony members for this goal, the agent starts a virtual dance which other colony members can observe. The dance communicates the directional information directly to the observers. (iv) agent state 'Rester' indicates the agent remains in the hive and is resting. It either did not find a goal by exploration or it did find a goal and recently stopped dancing for it. (v) agent state 'Observer' indicates an agent is looking for dancers to observe. Whenever it finds one, it determines whether the danced 'advertisement' is of high enough quality to adopt. (vi) agent state 'Recruit' indicates an agent adopted the directional information from a dancer and is traveling to the goal. Its actions are guided by the GV which is continuously updated during the travel to the goal. During the exploitation phase, the HV is also continuously updated.

$$b = \sqrt{a^2 + c^2 - 2ac \times cos\beta} \tag{1a}$$

$$\alpha = \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc}\right) \tag{1b}$$

Second, $CalculateVectors()$ is used to compute the PI vectors for each agent, i.e., the HV and possibly the GV. A PI vector essentially consists of two values, one indicating the direction and the other indicating the distance. A PI vector is always calculated with respect to the previous one. In order to calculate the new distance, we use the cosine rule and rewrite it to 1a. *a* represents the distance traveled since the last turn was made, *c* the old distance, and *b* the new distance. β is the angle turned with respect to the old angle. Using Equation 1a we can now calculate α (the angle used for adjusting the old angle), once again by using the cosine rule. Values obtained by Equation 1a and Equation 1b are used to construct the new PI vector.

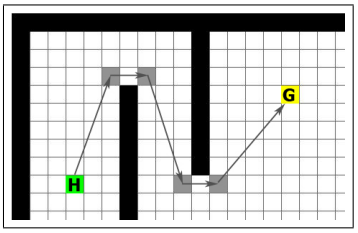


Figure 1: Basic example of a landmark network. 'H' and 'G' respectively represent the hive and goal. The grey squares represent the landmarks. The grey arrows represent a landmark route.

(a) Obstacle corner location

(b) Open corner location

(c) Corridor T location

(d) Corridor corner location

Figure 2: Situations in which a landmark ('L') is detected. Grey squares indicate possible obstacles, black squares indicate necessary obstacles. Constellations may be rotated and/or mirrored. (a) A corner of an obstacle. b) An open corner. c) A corridor with a T-shape. d) A 'corner' of a corridor. In none of these cases a hive or goal may be present.

Third, $DaemonActions()$ can be used to implement centralized actions which cannot be performed by single agents, such as collection of global information which can be used to decide whether it is useful to let an agent dance. In our bee-inspired algorithm, $DaemonActions()$ is not used.

3.2 The MAS landmark navigation algorithm

In order to improve the original bee-inspired algorithm, we extend it by adding landmark navigation. Essentially, this means that agents are able to create or represent sub-hives in the environment. In this research, agents create sub-hives in the environment and do not represent the landmarks by themselves. Each sub-hive contains directional information, in the form of HV's and/or GV's, to other sub-hives (or the goal) and form together a network of directed vectors from hive to goal and/or goal to hive. Figure 1 shows a basic example of such a sub-hive network. Because of the resemblance to biological landmarks, in the remainder of this paper we refer to these sub-hives as landmarks.

Each agent's memory is increased with three fields. First, the temporary vector (TV) represents the vector from the last known landmark (or hive) to the current landmark (or goal). On arrival at a landmark (or goal), the TV is reset. The TV is essentially the same as a local vector in biological landmark navigation [5]. Second, an agent is able to remember which route it is following. Each landmark can thus represent multiple routes towards the goal (and/or back to the hive). Third, an agent is able to remember a goal's quality.

The general algorithm structure is based on the original bee-inspired algorithm and thus consists of three functions, i.e., (i) $ManageBeeBehavior()$, (ii) $CalculateVectors()$, and (iii) $DeamonActions()$. With respect to these functions only the first one has a different implementation. The behavior of agents is determined by the six possible internal states. First, agent state 'Explorer' indicates the agent is exploring the environment for goals and performing Lévy flights to optimally cover the search space [23]. During exploration, the agent continuously updates its HV. While covering the search space, the agent is able to detect key locations in the environment which are then used to store the agents' TV. The landmark detection conditions can be found in Figures 2(a), 2(b), 2(c), and 2(d). Essentially, a key location is found on a corner of an obstacle, an open corner, a corridor in a T-shape, and in a corner of a corridor. During exploration, the TV is stored as a Landmark Homing Vector (LHV). Explorer Landmark creation/reinforcement conditions can be found in Table 1. These conditions are only valid if the TV is not directly (i.e., within the agent's view range) obstructed. If the TV is obstructed the exploring agent is not allowed to create any landmarks.

Second, agent state 'Carrier' indicates the agent found a goal and is returning an item from the goal to the hive. The actions an agent takes during the return are guided by the LHV of the landmarks the agent visits while following the route back home. Each visited landmark is also used to create/reinforce the Landmark

Table 1: Explorer/Carrier Landmark creation/reinforcement conditions. SL = Is State a Landmark?, ALR = Is Agent Landmark Route Known?, LVE = Landmark Vector Exists? I = 'Create new landmark with new agent landmark route', II = 'Create new landmark with known agent landmark route'.

SL	ALR	LVE	Explorer Consequence	Carrier Consequence
False	False	False	I	I
False	False	True	Agent adopts the vector's landmark route, reinforce vector	*Not possible*
False	True	False	II	II
False	True	True	II	II
True	False	False	*Not possible*	*Not possible*
True	False	True	*Not possible*	*Not possible*
True	True	False	II	if no LGV for agent route then set LGV with known agent route
True	True	True	if TV = LHV then reinforce vector	if Agent Goal Quality ¿ Landmark Goal Quality then overwrite LGV

Goal Vector (LGV). On arrival at a landmark, the agent stores its TV in the LGV and then resets the TV. Carrier Landmark creation/reinforcement conditions can be found in Table 1. If a carrier is not able to follow its route back (i.e., the LHV is blocked or the indicated follow-up landmark is non-existent) it returns to the last known landmark and starts exploring for landmarks. If it finds one it tries to adopt another landmark route home. The carrier is not allowed to create any LGV's if it altered its route. Third, 'Dancer' indicates that an agent arrived at the hive with directional knowledge of a landmark or possibly the goal directly. This knowledge is represented in the TV and the landmark route the agent followed. In order to recruit other colony members for this goal, the agent starts a virtual dance which other colony members can observe. The dance communicates the directional information and route directly to the observers. Fourth, agent state 'Rester' indicates the agent remains in the hive and is resting. It either did not find a goal by exploration or it did find a goal and recently stopped dancing for it. Fifth, agent state 'Observer' indicates an agent is looking for dancers to observe. Whenever it finds one, it determines whether the danced 'advertisement' is of high enough quality to adopt. Sixth, agent state 'Recruit' indicates an agent adopted the directional information from a dancer and is traveling to the goal using the landmarks' LGV on its route. The recruit essentially validates each LHV and LGV. More precisely, if a recruit is able to arrive at an indicated landmark then obviously the LGV is still usable. Likewise, the agent can compare its TV with the LHV. If the vectors are equal then obviously, the route back home is also still valid.

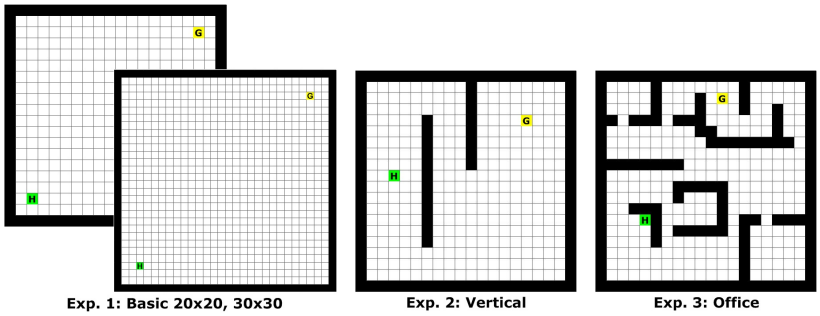
4 Experimental Set-up

In this section, we will discuss our experimental set-up, i.e., (i) the environments to experiment on, (ii) their characteristics, and (iii) the settings of the simulator. First, we perform simulation on three environments. Each environment consists of at least 20x20 cells and contains one hive and one goal, see Figure 3. We choose these four environments to show that the algorithm can deal with increasing environment complexity. Second, we describe the environment characteristics. In this research we use static environments. This means there are no goal and/or obstacle dynamics, i.e., all goals and obstacles remain locked in the same location. Furthermore, we assume all agents have perfect 'vision', i.e., in their view range, agents can perceive the quality of goals without errors.

EXPERIMENT 1: Basic. As a basic experiment, we run the simulation on two different-sized environments (i.e., 20x20 and 30x30 cells) without any obstacles between the hive and the goal. There is a hive in the lower-left corner and a goal in the upper-right corner. These experiments show how scalable the algorithms are with respect to world size. Moreover, we perform these experiments with 25 agents and 50 agents to show how scalable the algorithms are with respect to colony size.

EXPERIMENT 2: Vertical. To increase complexity, we add two vertical obstacles in the second experiment's environment. More precisely, we add two vertical obstacles and shift them with respect to each other. The hive is located at the middle left while the goal is located at the upper right of the environment. There is no direct path towards the goal or hive.

EXPERIMENT 3: Office. In order to further increase the complexity, we have constructed a maze-like



(a) Basic environments (b) Vertical environment (c) Office environment

Figure 3: Obstacle environments. 'H' and 'G' represent the hive and goal respectively.

environment which can represent an office. Again there is no direct path between hive and goal and there is a multitude of paths towards either of them.

Third, in order to get an indication about the performance of the landmark navigation algorithm, we compare the results to the performance of BSP [16] when applied to the same experiment set. We used the following settings for the simulators. (i) All agents start out as explorers, (ii) every agent that returns home as a carrier stays committed to its goal, (iii) the experiments are run with a colony size of 50 agents except in Experiment 1 which additionally is run with 25 agents, (iv) to limit run-time, we set a maximal time-step limit of 6000, (v) the maximal levy flight is set to 15 steps (i.e., cells to travel over), and (vi) to compensate for the inherent randomness in the algorithms, we perform each experiment 30 times.

5 Results

To have an indication of performance, we compare the results of our landmark navigation algorithm with the performance of BSP [16] applied to the same experiment set. We compare the performance of the algorithms with respect to their efficiency and scalability. Efficiency relates to the learning performance (i.e., knowledge acquisition and knowledge usage). If a MAS is efficient it will solve its task as fast as possible while using as few resources as possible. In this research, we measure efficiency by the average number of time steps the algorithm uses to complete the task. Moreover, by measuring the average computation time per time step, we also monitor time-related resources. Scalability relates to the ability to cope with an increasing problem size. A MAS' scalability is partly related to its efficiency. Obviously, an efficient MAS is more scalable than an inefficient MAS. To measure scalability in this research, we perform the experiments with different colony sizes and on two different-sized environments (i.e., Experiment 3(a)). In these experiments, a MAS that shows a small decrease in efficiency in an increasingly complex problem is more scalable than a MAS that shows a large decrease in efficiency. Moreover, if the computation-time ratio increases with increasing problem complexity, the MAS is less scalable than a MAS in which the computation-time ratio decreases. The computation-time ratio is calculated with $r = c_h/c_l$, where c_h represents the computation time in the high complexity case and c_l represents the computation time in the low complexity case.

Efficiency. We report our results with respect to efficiency in Table 2. We can observe that the landmark algorithm performs better on average. However, we can see that the standard deviation for BSP is low while for the landmark algorithm it is high. This is due to the nature of both algorithms. BSP constructs, with its inhibition pheromone, one single path between the hive and the goal. This naturally results in agents always taking the same path between hive and goal. The landmark algorithm however, constructs multiple paths which all have a probability of being used. In analogy to ant-inspired algorithms, the most efficient path gets selected over time. Note that the landmark algorithm is still faster at creating paths with respect to ant-inspired algorithms, due to the use of PI. The landmark algorithm does not have to wait until a strong enough pheromone trail emerges. Directional markers are immediately present. Path selecting is also more efficient due to the direct communication between agents. This in contrast with ant-inspired algorithms which have to look for pheromone trails strong and reliable enough to follow. By using a 2-sided Student's T-test with 95% confidence interval, we checked whether the difference between the efficiency performances of the two algorithms in Experiment 2 is significant. The test shows that the difference for experiment 2 is not significant (T2 = -1.5183) hence the landmark algorithm does not outperform BSP in Experiment 2.

Efficiency

Algorithm (# agents)	Experiment 1a		Experiment 1b		Experiment 2		Experiment 3	
	μ	σ	μ	σ	μ	σ	μ	σ
BSP (25)	1427.3	10.4	4323.1	30.5	-	-	-	-
BSP (50)	759.5	3.0	1937.0	9.2	1000.1	3.1	1910.1	12.9
LANDMARK (25)	957.3	53.4	1575.1	103.2	-	-	-	-
LANDMARK (50)	596.5	29.6	1009.6	57.1	1034.9	125.5	1507.3	458.7

Table 2: The efficiency of the two algorithms under study, i.e., the number of time steps used to complete the experiments at hand; both average and standard deviation are displayed. In every experiment, the maximum possible number is 6000.

Scalability

Algorithm (# agents)	Experiment 1a		Experiment 1b		Experiment 2		Experiment 3	
	μ	σ	μ	σ	μ	σ	μ	σ
BSP (25)	4.84	0.0	4.5	0.0	-	-	-	-
BSP (50)	9.4	0.0	8.8	0.0	9.1	0.0	9.1	0.0
LANDMARK (25)	15.4	10.7	24.4	15.2	-	-	-	-
LANDMARK (50)	37.5	17.3	59.3	28.0	40.3	18.3	49.2	24.0

Table 3: The scalability of the two algorithms, i.e., the required computation-time per time step in milliseconds; average and standard deviation are displayed.

This is due to BSP's nature which may construct a circular path for wall-following agents. The obstacle constellation in Experiment 2 does not allow such a circular path which results in an equal efficiency for both BSP and the landmark algorithm. Overall, we can conclude that the landmark navigation algorithm is more efficient in two of the three experiments.

Scalability. The results with respect to scalability can be found in Table 3. We can observe that BSP uses significantly less computation time per time step to perform the task. A low computation time per time step will effectively result in shorter run times for the algorithm. On average, BSP uses $\frac{1}{5}$th of the computation time per time step in comparison to the landmark algorithm' computation time per time step. With respect to the computation-time ratio, we can observe that the ratio for both algorithms stays equal when complexity is increased. For example, if we take 25 agents and increase the environment size, the ratio for BSP will be 0.9. If we take 50 agents and increase the environment size, the ratio for BSP will still be 0.9. Thus, computation time grows linearly with increasing complexity. Since this also holds for the landmark algorithm, we can conclude that both algorithms are equally scalable with respect to both environment size and colony size.

6 Conclusion And Future Work

In this research, we presented a novel landmark navigation algorithm based on a MAS. More precisely, the algorithm constructs stigmergic landmark networks over which agents can travel from and to a goal. In contrast to existing algorithms, we present an online learning approach in which found landmarks can immediately be used. Each landmark represents a segment of the total route, hereby ensuring robustness and accuracy by decreasing inherent accumulating PI errors.

Observing our results, we may conclude that the algorithm is at least as efficient as BSP. In two of the three experiments, the novel landmark navigation algorithm proves to be significant more efficient than BSP. However, its computation time per time step is five times larger which in the end results in longer running times for the experiments. However, the computation-time ratios for both algorithms do not change with increasing complexity by which we may conclude that computation time grows linearly with the complexity. Both algorithms are thus equally scalable. We believe, the landmark algorithm's computation time per time step can be improved by optimizing the algorithms code.

Future work is found in optimizing the algorithm for better (computation) performance. We also would like to know how the algorithm performs in dynamic environments, i.e., with goal and obstacle dynamics. In order to effectively test the robustness of the algorithm, we also want to include observation errors which add uncertainty about observed goal quality. The algorithm may be further improved by adding a cumulative quality/distance assessment which may prove a better route assessment strategy. It may also be interesting to measure the ratio between exploration and exploitation of the environment. And finally, we would like to introduce a measure for adaptability which obviously is a feature which is very important in dynamic environments.

References

[1] D'idac Busquets, Carles Sierra, and Ramon L'opez de M'antaras. A multiagent approach to qualitative landmark-based navigation. *Autonomous Robots*, 15:129–154, 2003.

[2] B.A. Cartwright and T.S. Collett. Landmark learning in bees. *Journal of Comparative Physiology A*, 151:521–543, 1983.

[3] K. Cheng, T.S. Collett, A. Pickhard, and R. Wehner. The use of visual landmarks by honeybees: bees weight landmarks according to their distance from the goal. *Journal of Comparative Physiology A*, 161:469–475, 1987.

[4] L. Chittka and J. Kunze. The significance of landmarks for path integration in homing honeybee foragers. *Naturwissenschaften*, 82:341–343, 1995.

[5] M. Collett, T.S. Collett, S. Bisch, and R. Wehner. Local and global vectors in desert ant navigation. *Nature*, 394:269–272, 1998.

[6] Matthew Collett, Duane Harland, and Thomas S. Collett. The use of landmarks and panoramic context in the performance of local vectors by navigating honeybees. *Journal of Experimental Biology*, 205:807–814, 2002.

[7] Thomas S. Collett and Matthew Collett. Memory use in insect visual navigation. *Nature*, 3:542–552, 2002.

[8] T.S. Collett. Insect navigation en route to the goal: multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 199:227–235, 1996.

[9] T.S. Collett, E. Dillmann, A. Ciger, and R. Wehner. Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, 170:435–442, 1992.

[10] J.L. Deneubourg, S. Aron, S. Goss, and J.M. Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behaviour*, 3:159–168, 1990.

[11] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.

[12] Matthias O. Franz, Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bülthoff. Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125, 1998.

[13] Matthias O. Franz, Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bülthoff. Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79:191–202, 1998.

[14] P. Gaussier, C. Joulain, J.P. Banquet, S. Leprtre, and A.Revel. The visual homing problem: An example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30:155–180, 2000.

[15] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64, 2000.

[16] Nyree Lemmens, Steven de Jong, Karl Tuyls, and Ann Nowé. Bee system with inhibition pheromones. In *European Conference on Complex Systems (ECCS)*, 2007.

[17] Nyree Lemmens, Steven de Jong, Karl Tuyls, and Ann Nowé. Bee behaviour in multi-agent systems: a bee foraging algorithm. *Lecture Notes in Artificial Intelligence: Adaptive Agents and MAS III*, LNAI 4865:145–156, 2008.

[18] Randolf Menzel, Karl Geiger, Lars Chittka, Jasdan Joerges, Jan Kunze, and Uli Müller. The knowledge base of bee navigation. *Journal of Experimental Biology*, 199:141–146, 1996.

[19] Ralf Möller. Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics*, 83:231–243, 2000.

[20] M. Müller and R. Wehner. Path integration in desert ants, *Cataglyphis Fortis*. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

[21] Carl Owen and Ulrich Nehmzow. Landmark-based navigation for a mobile robot. In *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (Complex Adaptive Systems)*, 1998.

[22] Kevin M. Passino and Thomas D. Seeley. Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off. *Behavioral Ecology and Sociobiology*, 59:427–442, 2006.

[23] M.F. van Dartel, E.O Postma, and H.J. van den Herik. Universal properties of adaptive behaviour. In H. Blockeel and M. Denecker, editors, *Proceedings of the 14th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'02), Leuven, Belgium*, pages 59–66, 2002.

[24] G. M. Viswanathan, V. Afanasyevc, Sergey V. Buldyrev, Shlomo Havlin, M. G. E. da Luze, E. P. Raposof, and H. Eugene Stanley. Lévy flights in random searches. *Physica A: Statistical Mechanics and its Applications*, 282:1–12, 2000.

[25] Anna N. Vlasak. Global and local spatial landmarks: their role during foraging by columbian ground squirrels (*Spermophilus columbianus*. *Animal Cognition*, 9:71–80, 2006.

[26] K. von Frisch. *The dance language and orientation of bees*. Harvard University Press, Cambridge, Massachusetts, 1967.

Distribute the Selfish Ambitions

Xiaoyu Mao [ab] Nico Roos [b] Alfons Salden [a]

[a] *Almende B.V.,*
Westerstraat 50, 3016 DJ Rotterdam, The Netherlands
[b] *MICC, Universiteit Maastricht,*
P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract

The distributed project scheduling problem is a distributed resource allocation problem extended with temporal constraints. Herein a project consists of a series of activities, the scarce resources are assigned to those activities at certain time windows such that all the activities are performed and certain objectives are optimized. In the past, methods from Operations Research (OR) and Artificial Intelligence (AI) were proposed to tackle this problem in a centralized manner. However, such centralized methods appear to be for many real-world applications impractical and infeasible due to the distribution of information, changes in willingness to cooperate and the decision autonomy of individual actors. Furthermore, long term schedules are likely to be invalidated by the dynamic changes in the environment. In this paper, we investigate this more dynamic problem in the context of Airport Ground Handling (AGH), where airlines and ground handlers are working together to deliver ground services during aircraft turnaround times. Although parties involved in AGH process cooperate, they may have their own selfish ambitions. For instance, in the scheduling process, airlines and ground service providers do not share a unified objective. In this paper, a multiagent online heuristic approach is proposed. We employ in this approach a market mechanism that balances the interests of airlines and ground service providers for scheduling the AGH services. The experimental results show that this multiagent approach provides a schedule that is of comparable quality as a centralized heuristic approach, and that by letting self-interested agents behave cooperatively, the overall schedule is significantly improved.

1 Introduction

The "on-ground" time of an aircraft at the airport starts with safe landing and finishes with next take-off. This period of time is called *turnaround* time in the aviation industry. During this turnaround time, a set of ground servicing tasks (e.g., passengers/baggage handling, aircraft maintenance, fuelling, cleaning and catering, etc.) must be performed. Ground handling involves very diverse tasks: some of them form a workflow (a sequence of services) with precedence constraints among each other, while others are independent service tasks. All such activities together form a so-called Activity-on-Node network [10]. Operations research defines scheduling the ground handling services of an individual aircraft as a *resource-constrained project scheduling problem* (RCPSP), in which the project refers to the aircraft turnaround and the activities to the ground services. The overall scheduling problem for the complete airport is refered to as Airport Ground Handling (AGH). As AGH deals with the scheduling of multiple aircraft ground services, this makes it a *resource-constrained **multi**-project scheduling problem* (RC**M**PSP) [3].

Both exact and heuristic centralized scheduling approaches to RCMPSP are proposed in OR and AI literature [8, 9, 11]. However, centralized approaches appear to be in many real-world applications impractical and infeasible when the decision has to be made by distributed and resource constrained actors. Assuming that these actors are independent economic units, it might be very well that these individual decision makers aim at optimizing their own objectives rather than the overall system performance. Such behaviours of actors call for project scheduling models and techniques that take the strategic behaviour of individual units into account [4]. For instance in AGH, ground handlers collaborate with airlines to deliver acceptable ground services during aircraft turnaround. On the one hand, during the process of scheduling ground service activities, airlines try to schedule their ground services as early as possible in order to minimize the turnaround

times - therewith to handle more flights a day. On the other hand, ground service providers which perform the ground services strive for as low as possible resource usage variation during the day, such that they can keep resource investments as low as possible.

The dynamic nature at an airport requires robust AGH schedules under various incidents. The actual arrival time of an aircraft is often different from the one on the original flight plan, either earlier or later caused by the uncertainties during the flight. Ground services may also take more time or less than expected. Under these dynamic features, the planned AGH schedule becomes obsolete, and requires re-scheduling or schedule-repair. An online scheduling approach reduces the risk of re-scheduling by starting the scheduling process of a project when it is released or the release time is approaching.

To acquire robust AGH schedules, which respect the selfish ambitions of distributed stakeholders, we propose a multiagent solution consisting of distributed autonomous agents representing aircraft (project agents) and service providers (resource agents), respectively, that either cooperate or not during scheduling of AGH servicing tasks by negotiating the slot price for these tasks. Herein the aircraft and service provider agents take flow time and resource usage into account. Finally, we will compare the performance of these multiagent scheduling approaches (non-cooperative and cooperative) with some well known centralized heuristic approaches.

The remainder of this paper is organized as follows. In Section 2, we define the problem of multi-project scheduling and discuss the objective functions to be optimized. Section 3 investigates the priority rule based heuristic methods for RCMPSP with different objectives. In Section 4 we present our distributed online heuristic approaches to solve RCMPSP based on slot-price negotiation amongst agents. Section 5 compares some experimental results obtained by our distributed agent heuristic method with the centralized OR heuristic methods. Section 6 concludes this work with a look to the future.

2 RCMPSP with different objectives

Many works published in the area of project scheduling make reference to the scheduling of a single project under resource limitations and time constraints. The scheduling problem we are investigating in AGH deals with multiple projects (i.e., aircraft turnarounds), mapping it to a RCMPSP [3].

A RCMPSP is a problem with M projects sharing K types of resources. Each project i consists of $j = 1, \ldots, N_i$ activities $a_{i,j}$ and has a release time t_i^r. A precedence constraint $a_{i,j} \prec a_{i,j'}$ may exist between two activities in a same project. Two fictitious activities $a_{i,0}$ and a_{i,N_i+1} are added for representing the "start" and the "end" of project i. Each activity $a_{i,j}$ requires a processing time $p_{i,j}$ and $r_{i,j}^k$ amount of resource of type $k \in K$ for its completion. Resource type k has a renewable capacity of R_k. The RCMPSP entails finding each activity a schedule $s_{i,j}$ (i.e., the start time) that forms an overall schedule $S = \{s_{i,j} \mid 1 \leq i \leq M, 1 \leq j \leq N_i\}$ satisfying both precedence and resource constraints and optimizing certain objectives.

Often, time-based objectives are the criteria to be optimized for RCMPSP. Most time-based objectives are defined as functions f that are nondecreasing in the start times of activities, so-called *regular objective functions* [10]. An example of such objective functions - also considered in this paper - is the *mean project flow time* [8, 9]:

$$f_{MPFT}(S) = \sum_{i=1}^{M} \frac{s_{i,N_i+1} - t_i^r}{M} \tag{1}$$

where the start time of the last fictitious activity $s_{i,N_i+1} \in S$ corresponds to the completion time of project i. Minimizing this criterion is equivalent to minimizing the *mean project delay time* when a due time is set for each project based on the length of its resource-unconstrained critical path.

However, non regular objective functions are needed in practice, for instance the resource-based objectives (e.g, *resource investment problem*, *resource levelling problem*, etc). Also in RCMPSP for AGH it is natural to investigate a resource levelling problem: a project scheduling problem where the objective is to minimize the resource usage variation over time. Although there are different measures for resource usage variation, we take the well known variation function called the *total squared utilization cost* [10]:

$$f_{TSUC}(S) = \sum_{k \in K} c_k \sum_{t=0} u_{t,k}^2 \tag{2}$$

in which resource utilization cost $c_k > 0$ is the cost incurred per unit of resource $k \in K$ per unit of time, and $u_{t,k}$ is the utilization of resource k scheduled at discrete time point t.

3 Centralized heuristic methods for RCMPSP

Realizing that most scheduling problems have to be done at a stage close to the practical operations, agility in decision making is the most important factor. Heuristics based on priority rules have been considered one of the most important solution techniques for large problem instances. These heuristics have the advantage of being intuitive, easy to implement, and fast due to low computational costs [1]. Here we present priority rule based heuristic approaches for RCMPSP with regard to two incompatible and potentially conflicting objective functions which are i) minimizing the *mean project flow time* (time-based objective) and ii) minimizing the *total squared utilization cost* (resource-based objective).

Priority rule based heuristic methods are employed in a so-called *schedule generation scheme* (SGS). In general, SGS starts from scratch and builds a feasible schedule by stepwise extension of a partial schedule. At each step (also called generation), two decisions have to be made: i) **selecting an activity** from the *eligible set*[1], ii) **deciding a start time** for this chosen activity. The different decisions made concerning these two issues result in different schedules. In priority rule based heuristic methods, priority rules are applied to help resolve the first issue of selecting an activity. Subsequently, the heuristic used in determining the start time of the chosen activity critically depends on the objective functions employed. We will recall in the following subsections the best performing heuristics that can be used in either selecting an activity or deciding a start time.

3.1 Selecting an activity

The priority rules are used in order to select an activity from the eligible set in each generation of SGS. In the priority rule based heuristic approach, eligible activities are sorted by a priority rule, and the activity with the highest priority will be scheduled next.

Priority rules for single project scheduling had been well studied in the past [5, 6]. When dealing with multiple projects, most of the priority rules for single project scheduling can still be used in an approach called single-project approach, which uses dummy activities and precedence arcs to combine the projects into a single mega-project, thereby reducing a RCMPSP to a RCPSP. On the contrary, a multi-project approach applies priority rules that take into account the project characteristics. The latter turned out to be more efficient when minimizing the time-based objective functions in RCMPSP [9]. Kurtulus and Davis [8] have for instance experimentally demonstrated the efficiency of rules like Maximum Total Work Content first (MAXTWK) and Shortest Activity from the Shortest Project first (SASP) in minimizing the *mean project flow time*.

Experiences by Neumann at al. [10] show that for *resource levelling problems* the combination of two priority rules, first using Minimum Slack Time (MST) followed by Greatest Resource Demand (GRD) performs best. We denote this combination as: MST+GRD.

3.2 Deciding a start time

When the objective is to minimize the *mean project flow time*, the objective function (1) is updated when an eligible activity $a_{i,j}$ is selected and added to a partial schedule S^c. We define a marginal cost function f_i^T, which describes the change in objective function as a result of scheduling an activity $a_{i,j}$:

$$f_i^T(S^c, s_{i,j}) = Est^{S^c \cup \{s_{i,j}\}}(a_{i,N_i+1}) - Est^{S^c}(a_{i,N_i+1}) \tag{3}$$

where $Est^{S^c}(a)$ denotes the lower bound of earliest precedence-feasible start time of activity a given the current partial schedule S^c. $Est^{S^c}(a_{i,N_i+1})$ is the lower bound of earliest start time of fictitious ending activity a_{i,N_i+1} and is therefore the earliest completion time of project i.

When SGS is used to solve project scheduling problem with such regular objective functions, deciding a start time in each SGS step becomes trivial - *choose the earliest feasible start time*. In this case, the priority rule(s) employed in selecting one activity from the eligible set will determine the final schedules.

[1] Eligible set: a set of activities that are not in the current partial schedule and whose predecessor activities have been scheduled.

When the objective functions to be minimized are non regular, choosing the start time of an activity therefore becomes crucial. Recall that in the resource levelling problem, total squared utilization cost function (2) is resource-dependent. When the schedule is constructed successively by SGS, this function is an accumulation of marginal costs added successively when a new activity is scheduled.

Suppose that the *current* utilization of resource k at time point t is $u^c_{t,k}$, and that an activity $a_{i,j}$ is to be scheduled at time $s_{i,j}$. Then the marginal resource cost of scheduling activity $a_{i,j}$ on resource k is:

$$f^R_{i,j}(S^c, s_{i,j}) = c_k \sum_{t=s_{i,j}}^{s_{i,j}+p_{i,j}} [(r^k_{i,j} + u^c_{t,k})^2 - (u^c_{t,k})^2] = c_k \sum_{t=s_{i,j}}^{s_{i,j}+p_{i,j}} [(r^k_{i,j})^2 + 2 \cdot r^k_{i,j} \cdot u^c_{t,k}] \quad (4)$$

Taking the example of Figure 1, we assume that the resource utilization price c_k is 1€. The activity lasting 4 time units with 2 resource units requirement and scheduled from time point 4 will have a marginal cost: $(7^2 - 5^2) + (8^2 - 6^2) + (6^2 - 4^2) + (4^2 - 2^2) = 84$€. Deciding the start time of a chosen activity is therefore to choose the greatest minimizer of this marginal cost.

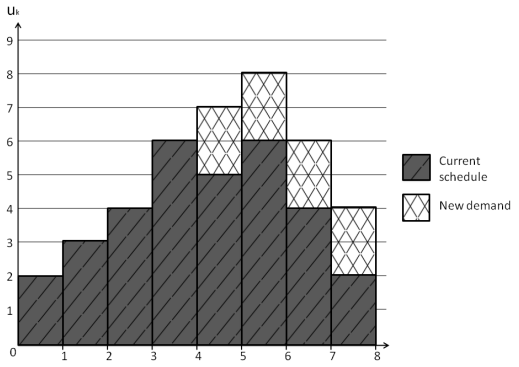


Figure 1: Example of resource usage chart

3.3 Combined objectives

In the real world applications, schedulers are often asked to schedule the projects with an overall goal that combines multiple objectives (i.e., multicriteria objective or multi-goal). For instance, combining mean project flow time and total squared resource utilization cost as our problem objective. Usually, a linear combination of the objective functions is used: $f = \alpha \cdot f' + (1 - \alpha) \cdot f''$. Another possibility is to associate costs to the flow-time of a project. By varying the delay cost, we can either emphasize minimizing the flow-time or optimizing the resource leveling.

Let dc_i denotes the delay cost per time unit for project i. Varying the value of dc_i can turn the problem emphasis from one objective to another. We therefore can define the combined problem objective as $f(S) = dc_i \cdot f_{MPFT} + f_{TSUC}$, and marginal cost function $f_{i,j}(S^c, s_{i,j}) = dc_i \cdot f^T_{i,j}(S^c, s_{i,j}) + f^R_{i,j}(S^c, s_{i,j})$. The relation between the variable dc_i and α is deducted: $dc_i = \frac{\alpha}{1-\alpha}$.

4 Multiagent online heuristics

We discussed in the previous section the traditional project scheduling heuristics using a centralized authority to solve an optimization problem. This central decision maker would be equipped with all relevant data of that problem, asked to derive a solution that fulfills all the necessary side constraints, and optimize some kind of global performance criteria, either time-based or resource-based, simultaneously. In decentralized settings, central coordination of a system is replaced by decisions and actions locally taken by distributed computational agents. Those agents are assumed to act rationally with respect to their own interests, they communicate with each other in order to make local decisions and eventually a global schedule will emerge.

Multi-project scheduling problems are often studied in a deterministic environment, where the project release time, the activity processing time are all static. In a dynamic environment like AGH, aircraft may arrive earlier or later than the original plan (uncertainty of release time), and ground service may take longer or shorter than expected (uncertainty of processing time). To uncertainties in project release time,

we propose an online version of a distributed heuristic. Instead of scheduling all projects at a time, we let agents start making schedules when the project is released.

Two types of agents have to be defined in the case of a multiagent project scheduling problem: ***project agents*** (PAs) and ***resource agents*** (RAs). Each project agent, e.g. an aircraft or airline company in AGH, are responsible for a project consisting of a set of activities, while information about internal precedence constraints among activities of other project agents are hidden for them. Activities in these projects need certain resources provided by the resource agents for their completion. Each resource agent is responsible for a resource type; it receives the resource requirement request concerning a certain activity from a project agent; and it negotiates with the latter project agent the start time of this activity. Agent negotiation is carried out by adopting a slot purchasing market mechanism.

4.1 Slot-price market mechanism

Communication between agents occurs when a resource agent RA_k receives a resource requirement request concerning activity $a_{i,j}$ sent by project agent PA_i. In order to complete its project as early as possible, PA_i prefers this activity to be scheduled at the earliest time-slot possible. However, this earliest time-slot for $a_{i,j}$ to be carried out on behalf of agent PA_i may not be an appropriate slot for RA_k: it may conflict with the objective of RA_k of minimizing the variation of its resource utilizations over time.

To resolve this conflict, we employ a market mechanism that is a natural way of letting parties with different interests and goals appreciate goods or services. The price of each time-slot is determined by agent RA_k, on the basis of the agent's internal value system (in the resource levelling case, this price would be calculated by the marginal cost function (4)). Resource agents assign higher prices to popular time-slots requested by late coming project agents. By using this pricing model resource agents can force project agents to choose a cheaper slot in order to reduce their own resource levelling costs. Project agents, choosing a later (maybe cheaper) slot might incur a higher delay cost. The delay cost incurred by scheduling can be expressed in the marginal function (3). The actual price of one slot for project agent PA_i is therefore the addition of price given by resource agent RA_k and the delay cost incurred by a late schedule.

Now the negotiation amongst project agent PA_i and resource agent RA_k to optimize both delay and resource usage costs simultaneously goes as follows:

1. When project agent PA_i is released, it selects one of its eligible activities $a_{i,j}$ to be scheduled and sends the resource requirement (including the earliest start time $EST_{i,j}$, the latest start time $LST_{i,j}$, $p_{i,j}$ and $r^k_{i,j}$) to the corresponding resource agent RA_k;
2. RA_k by receiving this request uses its pricing model (i.e., function (4)) on the resource requirements to calculate a list of offers, starting from the earliest possible slot; it then sends this list of offers to project agent PA_i;
3. PA_i receives this list of offers, adds the possible delay costs to these slots, selects the cheapest slot for scheduling activity $a_{i,j}$ and informs resource agent RA_k about its selection.
4. RA_k will acknowledge this decision, and schedules $a_{i,j}$ on this selected slot, and keeps listening to the new upcoming requesting project agents.
5. PA_i will move on to schedule the next eligible activity.

With this agent negotiation model, we successfully distribute the decision making responsibilities and concerns to individual and different types of agents, and the overall schedule of all projects emerges.

4.2 Cooperative agent scheduling

Instead that both project agent and resource agent keep their personal information private, we propose in this section a cooperative agent scheduling scheme: project agents are willing to share part of their personal information to resource agents, under the condition that they don't suffer any loss from information sharing. Being cooperative is not contradictory to the selfish nature of individual agents, instead, it has been proved to be beneficial not only for common goods but also for the selfish individuals in a non zero-sum game [2], as in AGH.

Let us illustrate this type of agent collaboration. Figure 2 presents a time line of resource type k, assuming that resource agent RA_k managing such resource type. Activity B with $r_B = 2$ and $p_B = 5$ has been scheduled on this resource at the time window $[3, 8)$, the current resource levelling value is $5 \times 2^2 = 20$. A newly arrived activity A ($r_A = 2, p_A = 4$) is requesting the slot price starting from 4. According to the pricing model of resource agents presented earlier, the slot price of $[4, 8)$ is: $2^2 + 4 \times (4^2) - 20 = 48$ (cf. Figure 2, option 1).

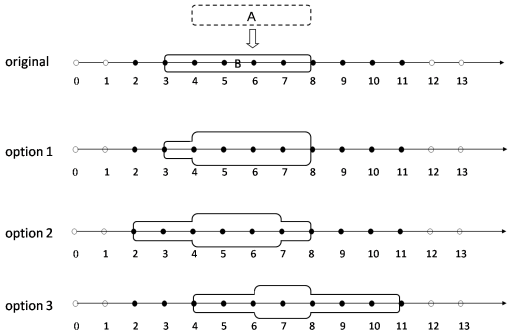


Figure 2: Activity A to be scheduled on a resource

In the cooperative scheduling scheme, project agent PA_B which manages activity B is willing to reveal some additional information to resource agent RA_k. For instance, when all the direct successors of activity B are scheduled (might be on other resources), activity B will have a *secure time window* - starting from the latest scheduled finishing time of its predecessors, and ending by the earliest scheduled starting time of its successors. Any shifting within this secure time window of B will not cause any further delay of the project that B belongs to. By knowing the secure time windows of activities, resource agents RAs are given more flexibilities to shift the slots for decreasing the resource levelling value.

In Figure 2, the black dots on the resource time line are indicating the secure time window of activity B. Option 2 in Figure 2 illustrates the case where B is shifted backward to $[2, 7)$ first, and A is scheduled on $[4, 8)$, this shifting will result in a lower total resource levelling: $3 \times 2^2 + 3 \times 4^2 = 60$, therefore, the marginal cost of scheduling activity A is only: $60 - 20 = 40$. Instead of shifting B backward, in Figure 2 option 3, B is shifted forward to $[6, 11)$, which gives an even lower marginal cost of scheduling activity A: $5 \times 2^2 + 2 \times 4^2 - 20 = 32$.

In this cooperative agent scheduling scheme, by providing more information to resource agents, project agents are not suffering any loss but helping resource agents for a lower levelling cost. Subsequently, cheaper slot options are provided to other project agents.

5 Experimental Results

We conducted the experiments on two types of data set: 1) a multi-project instance built from the standard benchmark data set $J30$ of the Project Scheduling Problem Library (PSPLib) [7]; 2) a simulated airport specialized multi-project data set. Both experiments are carried out with three centralized priority-rule heuristic approaches (MST+GRD, MAXTWK, SASP) and two multiagent approaches (Non-cooperative, Cooperative).

In the centralized heuristic methods, the differences in three methods are the priority rule(s) applied in selecting an eligible activity. For deciding the activity start times, they all use the same heuristic - choose the start time resulting in the minimum combined marginal cost function $f_{i,j}(S^c, s_{i,j}) = dc_i \cdot f^T_{i,j}(S^c, s_{i,j}) + f^R_{i,j}(S^c, s_{i,j})$. By decreasing gradually the value of dc_i, the problem will turn from a time-optimization problem (minimizing *mean project flow time*) to a resource-optimization problem (minimizing *total resource utilization cost*).

In multiagent heuristic methods, resource agents are assumed using the same pricing model (i.e., marginal resource cost function (4)). We achieve the emphasis turning from reducing *mean project flow time* to *resource levelling* by gradually decreasing the delay cost per time units dc_i. Recall that, project agents are concerned about the delay incurred by scheduling an activity. Choosing a high dc_i value emphasizes time-based objective, on the contrary, a low dc_i turns the problem emphasis to resource-based objective.

Experiments for benchmark instances

In the first data set, we combine 10 project instances of single-mode projects in PSPLib ($J301\_1 \rightarrow J301\_10$) into one multi-project problem. Each of the projects is composed of 30 real activities with precedence dependencies plus a dummy start and a dummy end activity. 4 types of renewable resources exist

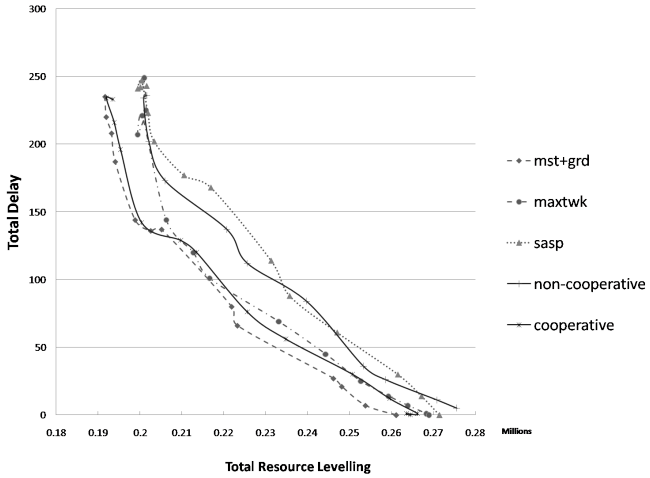


Figure 3: Various approaches in solving benchmark problem with bi-criteria objectives

and, each of which is required for executing an activity. The capacity of each resource type is the summed resource capacity in each instance. Projects are released sequentially, in a pace of every 5 time units.

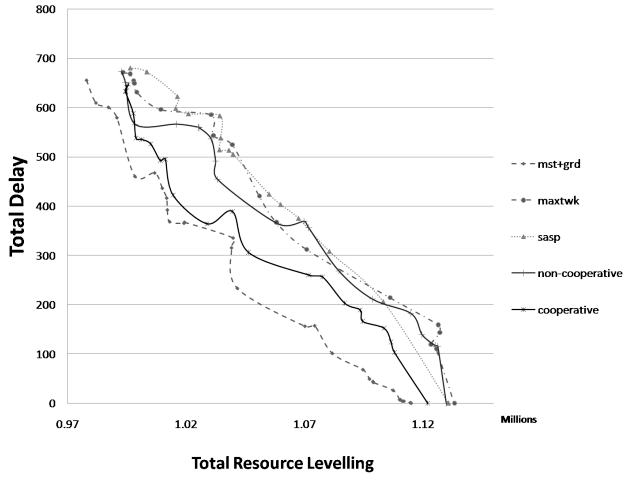


Figure 4: Various approaches in solving AGH problem with bi-criteria objectives

Experimental results in Figure 3 show that i) in all solution methods, the higher resource levelling value goes (cf. x-axis), the less total delay (cf. y-axis: equivalent as *mean project flow times* in representing time-based objective) will be suffered; ii) Non-cooperative online multiagent heuristic approach provides a schedule that is of comparable quality of centralized heuristic methods. iii) Moreover, by letting self-interested agents behave cooperative, given the same total delay, we can have lower resource levelling value. Therefore, the overall schedule made by cooperative agent scheduling heuristic is significantly improved: it is comparable with the best heuristic methods.

Experiments for airport specialized instances

An instance of AGH multi-project scheduling problem has the features that all projects have similar network structure and similar activity set. In order to build an AGH multi-project scheduling problem instance, we simulate a multi-project set by cloning a single-project instance in PSPLib. We conduct the experiments with 30 identical projects created out of the project instance $J301\_1$, each project represents an aircraft turnaround, and aircraft are released every 5 time units. As in the benchmark problem experiments, five

solution methods are investigated in this AGH problem, Experimental results in Figure 4 show similar results in which the cooperative agent approach outperforms two of the centralized heuristic approaches.

6 Conclusions & Future Work

This paper investigated several centralized priority-rule based heuristic approaches for solving multi-project scheduling problems with different objectives. We discuss the futility of these approaches when autonomous decision making is unavoidable, because the stakeholders involved are not willing to share their scheduling or pricing strategies as in the case of AGH. In order to tackle this distributed multi-project problem under release uncertainties, we developed an on online multiagent heuristic approach, and extended this approach with an agent cooperative scheduling scheme.

The experimental results show that our online multiagent heuristic method generates schedules that are of comparable quality as centralized heuristic methods. By letting self-interested agents behave cooperatively — share non-critical information — the overall schedule is significantly improved in the sense that it is of comparable quality as the best heuristic methods.

To schedule activities online is to be able to cope with a dynamically changing environment, in future work we will investigate the robustness of these online multiagent approaches (non-cooperative and cooperative) in the context of uncertain activity processing duration. Other research directions such as learning, coalition formation and other marketing mechanisms will be also pursued.

References

[1] P. Brucker, A. Drexl, R. Mohring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112:3–41, 1 January 1999.

[2] R. Dawkins. *The Selfish Gene: 30th Anniversary Edition — with a New Introduction by the Author*. Oxford University Press, 2006.

[3] U. Dorndorf. *Project Scheduling with Time Windows: From Theory to Applications*. Physica-Verlag, 2002.

[4] B. Heydenreich, R. Müller, and M. Uetz. Decentralization and mechanism design for online machine scheduling. In *Algorithm Theory - SWAT 2006*, volume 4059 of *Lecture Notes in Computer Science*, pages 136–147. Springer, 2006.

[5] R. Kolisch and S. Hartmann. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In J. Weglarz, editor, *Project scheduling: Recent models, algorithms and applications*, pages 147 – 178. Kluwer academic, 1999.

[6] R. Kolisch and S. Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1):23–37, 2006.

[7] R. Kolisch and A. Sprecher. Psplib — a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1997.

[8] I.S. Kurtulus and E.W. Davis. Multi-project scheduling: Categorization of heuristic rules performance. *Management Science*, 28(2):161–172, 1982.

[9] A. Lova and P. Tormos. Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. *Annals of Operations Research*, 102(1-4):263–286, February 2001.

[10] K. Neumann, C. Schwindt, and J. Zimmermann. *Project Scheduling with Time Windows and Scarce Resources*. Springer, 2nd edition, 2001.

[11] A. Yassine, C. Meier, and T. Browning. Multi-project scheduling using competent genetic algorithms. Technical report, University of Illinois, Department of Industrial & Enterprise Systems Engineering (IESE), 2007.

Closing the Information Loop

Jan Willem Marck [a] Sicco Pier van Gosliga [ab]

[a] *TNO Defence, Safety and Security, P.O.Box 96864, 2509 JG The Hague*
[b] *Intelligent Systems Lab Amsterdam, University of Amsterdam, The Netherlands*

Abstract

Decision support systems require a form of situation awareness. To generate situation awareness information is needed. Not all available information is necessary or equally influential. This paper proposes a way to determine *which* and *when* information is relevant. The goal of this is to minimize communication and processing of irrelevant information.

Our system is inspired by a few first responder experiments done in our lab. In these experiments first responders had to respond to a calamity. The information need of responders was analyzed. To have good team performance it was clear that at certain times certain information was important. We modeled a toy problem after this scenario and we use this illustrate our method of reducing irrelevant information. Our toy problem consists of a Bayesian network with which sensitivity analysis is used to illustrate *which* information is relevant. A simple tracker scenario with information theoretic techniques is used to illustrate *when* information is relevant.

1 Introduction

Sensor information is getting cheaper, richer and more accurate. As more sensors and information fusion processes are connected to decision support systems, the more challenging orchestrating the information flow between the functional components becomes. Various types of sensors have limitations (e.g. field of view), share resources (processing, bandwidth and energy) or simply interfere with each other. Different information fusion processes need and provide information in different modalities. In various contexts the end-users need different sets of information. The challenge lies in finding an effective way to get a common agreement on the priority of information between system modules that work on different levels of abstraction. Where sensor processes know the cost of gathering information, they can not independently decide on what information is relevant to higher processes.

We propose a concept for an intelligent loop between two functional components (FC). The purpose of this loop is to provide the relevant information to requesting FC and to aid the answering FC in determining the priority/relevance of it's answer.

We will use a field test of our first responder emergency staff at TNO Defense, Security and Safety in The Hague as the leading example. At this experiment our in-house emergency staff was supplied with PDA's connected wirelessly to a heterogeneous sensor palette. During a simulated emergency situation we investigated how the information demand of the first responder interacted with the supplied information. Inspired by the experimental feedback we have developed a concept for generating situational awareness combined with sensor management.

1.1 The concept

Our proposed concept comprises two communicating components. For this paper we choose two components in different levels of the JDL [7] information fusion model. The first component is a process capable of fusing object information upon request (JDL level 1). The information generated by this process is communicated to the second component, a Bayesian Network (JDL level 2) where the observations of sensors are used for an assessment of the situation. The concept is depicted in Fig. 1(a).

The goal of an object refinement process is to provide object information to situational assessment processes by combining sensor data. The goal of a situational assessment processes is to provide a good situation description for the impact assessment level. This is done by combining information from object refinement and a priori knowledge, e.g. intelligence information into a description of relations between all objects. This continuous loop of requesting and providing information is depicted Fig. 1(a).

The standard method of subscribing to a provider component results in communication whenever there is new information, usually this results in fixed update rates. In our concept the component that provides the information communicates when the information is relevant to the receiver. To determine the relevance of the information the transmitting component uses a certain divergence measure between communication the information or not. If this exceeds a certain threshold it is considered relevant and it is communicated. This results in behaviour that communicates a lot when information is relevant and less when it is less important.

First we will describe a method to determine *which* information is relevant. After that we will describe the method for *when* information is relevant.

2 Bayesian model

To process the information from object information to situation related information we use a Bayesian belief network [8]. Bayesian belief networks have become a popular framework for consistent reasoning with uncertainty. The variables of interest are connected as nodes in a directed acyclic graph. A directed arc represents a conditional dependency between a parent node and a child node. All variables are discrete, i.e. have a finite set of possible states. Each variable is accompanied with a conditional probability table (CPT) to describe how a child is influenced by the state of its parents.

In the Bayesian network shown in figure 1(b) we discriminate three abstract layers in the model to group together variables of a specific kind: a *input layer*, an *intermediate hypothesis layer* and an *output layer*. The input layer receives information from the object refinement process related to one or more variables in this layer ($S_{1..6}$). This information is used as evidence in the belief network. These observations may be accompanied with a probability to express the certainty of the observation. We use an intermediate hypothesis layer to interpret the observations by sensors on a more abstract level, so that they may be used in the output level. These variables describe phenomena like the presence of smoke, fire and people. The output layer has multiple hypotheses ($H_{1..4}$) available as argumentation for a decision. For example, in figure 1(b) the output layer includes variables for the safety of evacuation routes. In our first responder example the results of the output layer are used in a decision support tool.

Bayes rule is used to determine the posterior probabilities of the hypothesis H_x, given evidence e by sensors. Evidence e is a set of known values for variables in the sensor layer. The generalized version of Bayes rule is:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{1}$$

The certainty at which sensor x observes some phenomena A when it measures S_x is encoded in the prior conditional probability table $P(S_x|A)$, present at the sensor layer. Optionally, this table may include additional conditioning variables that affect the certainty of measuring A: $P(S_x|A, C, .., D)$. Bayes rule is applied to update the posterior beliefs for each variable in Fig. 1(b). This process is called belief updating and is NP-hard [1]. The details of deriving the posteriors of hypothesis $H_{1..4}$ given evidence at $S_{1..6}$ will not be discussed in this article. Various methods exist to efficiently apply Eq. 1 for calculating all posteriors. The books of Jensen [4] and Pearl [8] contain elaborated descriptions of such methods.

3 Sensitivity analysis

Sensors make observations. These observations are fused by processes in JDL level 1 (object refinement). These processes can not independently decide on how relevant information is for the given situation, since the situation is assessed at JDL 2 and above. To prevent the fusion of irrelevant information, relevance feedback is required from the process that uses the information.

The Bayesian model presented in Section 2 derives the posteriors of hypothesis H_x given evidence e concerning all sensor information at $S_{1..6}$. This requires the communication of beliefs; i.e. probabilities.

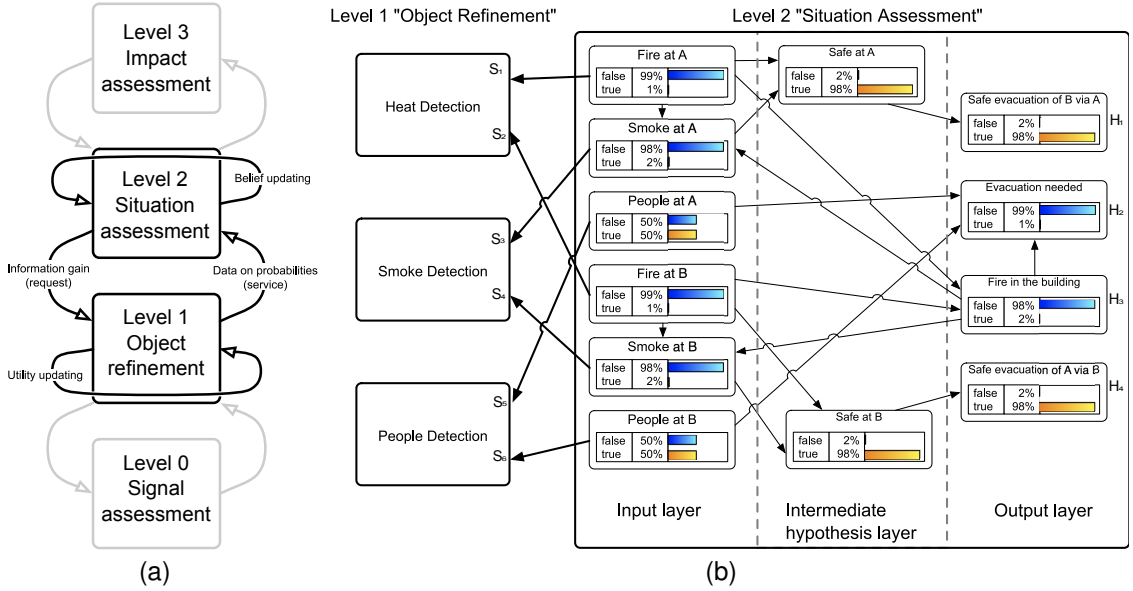


Figure 1: The proposed method closes the loop between JDL information fusion levels 1 and 2. (a) Overview of the components of JDL level 1 and the level 2. (b) Bayesian network describing variables of interest concerning emergency management for a building.

S_x	H_1 safe at A		H_2 evacuate		H_3 fire		H_4 safe at B	
	$G_1(S_x)$	$G_r(S_x)$	$G_2(S_x)$	$G_r(S_x)$	$G_3(S_x)$	$G_r(S_x)$	$G_4(S_x)$	$G_r(S_x)$
S_1 heat at A	0.711	0.935	0.445	0.519	0.733	0.959	0.177	0.201
S_2 heat at B	0.170	0.193	0.446	0.519	0.735	0.959	0.729	0.953
S_3 smoke at A	0.166	0.696	0.062	0.197	0.085	0.282	0.025	0.085
S_4 smoke at B	0.034	0.152	0.031	0.121	0.042	0.169	0.120	0.570
S_5 people at A	0	0.696	0	0.197	0	0.282	0	0.085
S_6 people at B	0	0	0.001	0	0	0	0	0

Table 1: The relevance of information by each sensor for each hypothesis. Where the value is zero S_x is irrelevant.

This same Bayesian framework can be used to determine the relevance of sensor information for any sensor S_y, as a subset of all observations $S_y \in e$.

Sensitivity analysis (SA) is a method to assess the effects of changing parameters in a model [2]. For a belief network SA measures the influence of a variables CPT on the posterior probabilities of others. This may be achieved by one-way sensitivity analysis in which the parameters of a variable are varied one at a time, while the posteriors of others are compared by the original unaltered results. A common task for SA is to identify the most influential parts of a belief network as an aid for modeling prior knowledge. Sensitivity analysis may be an intractable task, when a belief network is extensively tested for all possible variations.

However, in our case it is sufficient to use all possible instantiations of only sensor variable S_y as simulated evidence to evaluate the impact on a specific hypothesis H_x in the decision layer. This concerns altering the marginal distribution $P(S_y)$ in Fig. 1(b) instead of all priors in $P(S_y|H_y)$. Evaluation by simulation in a belief network is called *what–if analysis*. The cost of a single simulation is similar to normal inference. The model in Fig. 1(b) requires only two simulation runs per sensor, one for each state. Therefore SA will remain tractable for this application.

There a multiple alternative means to measure the effect on a hypothesis in the decision layer. The absolute difference in posterior distributions can not be used, because it gives misleading results: an absolute difference between 0% and 10% means something quite different than the same difference between 45% and 55%. The Jensen-Shannon (J-S) divergence [5] is a more appropriate way to evaluate the difference in posterior probability distributions. The generalized form of the J-S divergence is:

$$JSD(A \parallel B) = \frac{1}{2}KL(A \parallel M) + \frac{1}{2}KL(B \parallel M) \tag{2}$$

Where $M = \frac{1}{2}(A + B)$ and $KL(A \parallel B)$ is the Kullback-Leibler divergence between A and B. The following function G_y is used to assess the relative entropy of observing S_x for hypothesis H_y, where: $c = e \setminus S_x$:

$$G_y(S_x) = max\left(JSD\left(P(H_y|c) \parallel P(H_y|c, S_x)\right), JSD\left(P(H_y|c) \parallel P(H_y|c, \neg S_x)\right)\right) \tag{3}$$

This function evaluates the gain of information when sensor x is used for the assessment of H_y, taking into account the current observation of (S_x) as well as other possible observations $(\neg S_x)$. To express the information gain of observing S_x we take posteriors for hypothesis H_y given S_x and $\neg S_x$, and return the maximum J-S divergence from the posteriors of H_y without evidence for S_x. This is shown in Table 1 on the left side of each column.

A difficulty of using Eq. 3 to derive G_y for S_x, is that Bayesian inference in general makes up for missing information by exploiting the redundancy in sensor information. Therefore, the influence of missing or changing the outcome of just a single sensor is relatively low. To prevent underestimating the impact of a sensor, one should change one sensor at a time and recalculate G_y for every sensor. For example, after disabling S_4 on the basis its low information gain in Table 1, the observations of S_3 will become relevant. The sensor receives a request (G_r) to comply to the information gain of the parent of S_x (e.g. smoke or fire) for H_y, see Eq. 4.

$$G_r(S_x) = G_y(parentsOf(S_x)) \tag{4}$$

4 Relevant Communication

In sensor data fusion (e.g. tracking) results are rated by absolute criteria like track continuity, percentage of detected targets or length of traffic jam. These processes are at the moment not coördinated by shifts in importance between several of these kind of goals. E.g. in our first responder scenario the goal is to comply with the information need of our emergency staff, which changes as the situation changes.

In this scenario quick changes of situations occur. When a (simulated) fire incident happens the first responders have a multistage action plan. At first they have to localize the incident then they have to consider the implications of the incident and follow protocol accordingly. In each stage the first responders need different information. Information requests to the decision support system range from "Where is the fire?" to "Is the building evacuated?". At different times different hypotheses are important.

In our experimental setting we have a sensor palette (cameras, smoke detectors, indoor localization) able to provide information to the Bayesian network. An example in Fig. 2 illustrates the situation. Process A, in this case the Bayesian network, communicates with different processes/sensors to requests it information. Sensitivity Analysis of the Bayesian network indicates the relative entropy for each unknown variable for when that variable changes from unknown to known, in other words what the impact is of the sensor information to the Bayesian Network. If the divergence is low the information is not relevant and communicating this information to the Bayesian Network will not make a large difference. If the divergence is big the information might alter the state significantly so it should be communicated to the Bayesian Network.

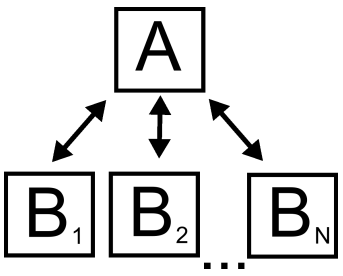


Figure 2: Process A requests data from multiple processes B_i to get information. Sensitivity Analysis indicates which process supplies relevant information. Relevancy Analysis indicates when this information should be transmitted.

Sensitivity Analysis provides a way to determine *which* information is relevant or irrelevant for a given situation. After determining which information is relevant enough to the Bayesian Network to be communicated, the interaction between the Bayesian Network and the information provider can be optimized as well. Various properties determine the effectiveness of the interaction: Aspects like bandwidth, update rate, dealing with limited or costly resources, etc... All these properties are to some extent negatively influenced by transmission and processing of irrelevant information. The following sections deal with a novel technique to determine *when* to transmit the relevant information [6].

4.1 Relevance

To determine *when* information is communicated it is important to determine when information is relevant. To determine the difference between communicating or not the Kullback Leibler Divergence (KLD) (Eq. 5 and Eq. 6) [3] is used. Eq. 7 describes the divergence of the consequences of the choice of incorporating (and transmitting) sensor information Z_{B_i}. In other words will Z_{B_i} have impact on state X_A.

$$D = D_{KL}(P||Q) \tag{5}$$

$$D = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{6}$$

$$D_i = D_{KL}(X_A|Z_{B_i}||X_A) \tag{7}$$

In which Z_{B_i} is the information in process B_i. X_A is the state of process A. In this case the hypotheses in the Bayesian Network in which the user is interested. D_i is the divergence between X_A and $X_A|Z_{B_i}$. That is, the difference between incorporating Z_{B_i} into X_A or not.

To calculate this KLD, knowledge of process A and state X_A and Z_{B_i} is necessary in one process. But both processes B_i and A contain only part of the necessary information to calculate the divergence. Process A has up to date knowledge of state X_A and has knowledge about its own process. Process B_i has the up to date knowledge of Z_{B_i} and should determine when to send this to A.

It is undesired for B_i to calculate the divergence because this requires intricate knowledge of process A and state X_A in B_i. This can be done by giving B_i the functionality of A. But this will turn the system into a monolithic construct and not a scalable, adaptive and modular system. Besides if there is one such process B_i, A will be irrelevant. If A needs multiple B_i processes to fulfill its task, placing the functionality of A in every B_i will not solve the problem, because it still needs information from other B_i processes.

It is clearly not possible for process A to determine the divergence before Z_{B_i} is sent to A.

The remaining possibility is that A and B_i calculate the divergence partially. The KLD is a function of X_A and Z_{B_i}. X_A could be marginalized out of the KLD. This remaining is a function over Z_{B_i}. Process A could calculate this marginal and send this over to B_i. B_i finishes calculates the divergence for a current Z_{B_I} to see if it results in a big divergence. If so, Z_{B_I} has impact in state X_A and Z_{B_I} is sent from B_i to A.

4.2 Math

Eq. 7 defines D_i as the KLD over $P(X_A|Z_{B_i})$ and $P(X_A)$. This can be rewritten into Eq. 8.

$$D_i = \sum_{z \in Z_{B_i}} \sum_{x \in X_A} P(x|z) \log \frac{P(x|z)}{P(x)} \tag{8}$$

D_i is the KLD in process A caused by the knowledge information Z_{B_i} . Z_{B_i} is the information requested by process A. Z_{B_i} is only known in process B_i. B_i should decide when to transmit this to A. X_A is the state of process A. X_A only known in process A.

Eq. 8 is rewritten into Eq. 9 by turning the conditional $P(x|z)$ using the Bayes Theorem and splitting the log. All factors that depend on only z are moved out of sum over x.

$$\begin{aligned} D_i = & \sum_{z \in Z_{B_i}} \frac{1}{P(z)} \sum_{x \in X_A} P(z|x)P(x) \log P(z|x) - \\ & \sum_{z \in Z_{B_i}} \frac{\log P(z)}{P(z)} \sum_{x \in X_A} P(z|x)P(x) \end{aligned} \tag{9}$$

The sums over X_A are the marginal that should be calculated in A and subsequently be sent to B_i, so B_i can finish calculating D_i. These sums are E_i and F_i:

$$E_i(z) = \sum_{x \in X_A} P(z|x)P(x)\log P(z|x) \tag{10}$$

$$F_i(z) = \sum_{x \in X_A} P(z|x)P(x) \tag{11}$$

By knowing E_i and F_i process B_i can now calculate D_i using Eq. 12 and determine when to send an update to A.

$$D_i = \sum_{z \in Z_{B_i}} \frac{E_i(z) - F_i(z)\log P(z)}{P(z)} \tag{12}$$

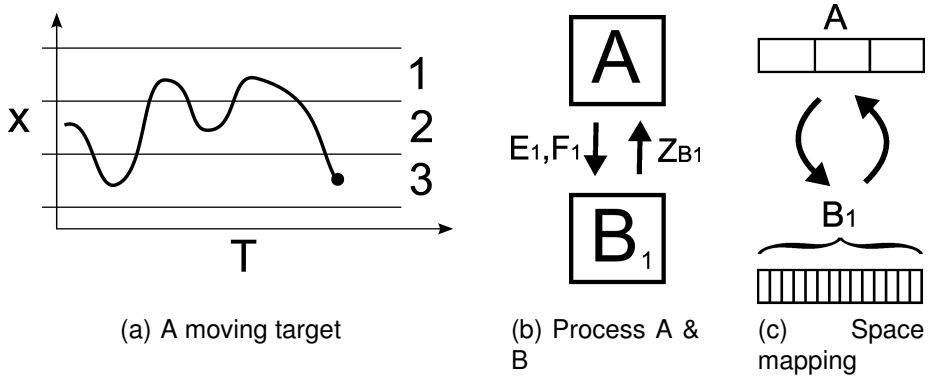
4.2.1 Timing

The decision to send Z_{B_i} is made by process B_i using E_i, F_i and the current information it is holding, Z_{B_i}. A proper send decision can only be made if E_i and F_i are up to date. The following is a good heuristic to keep E_i and F_i up to date without creating too much communication overhead.

As soon as process A gets information Z_{B_i} from process B_i it calculates E_i and F_i and sends these to B_i. B_i uses these to determine when to send the next update. As soon as the next update arrives at A, A recalculates E_i and F_i and sends them to B_i and so forth.

4.3 Example

To illustrate the method an example is given (Fig. 3). For example process A calculates the probability that a target is in a specified area. Process B_i can deliver target position information to other processes. The method discussed above is used to have relevant communication between A and B. In other words B transmits a position update to A as long as it results in a change in the state of A. Results are given in Fig. 4.



(a) A moving target (b) Process A & B (c) Space mapping

Figure 3: Process A requests information from a process B_1. Process B_1 provides position information to process A, which needs to give accurate picture whether the target is in an area.

5 Conclusion

We have chosen to use probabilities and the Kullback Leibler Divergence to determine relevance of communication between different functional components. We propose a method that enables efficient communication between these components in a generic way and this is illustrated in simple toy problems. The preliminary results are encouraging and indicate that this method should be investigated more.

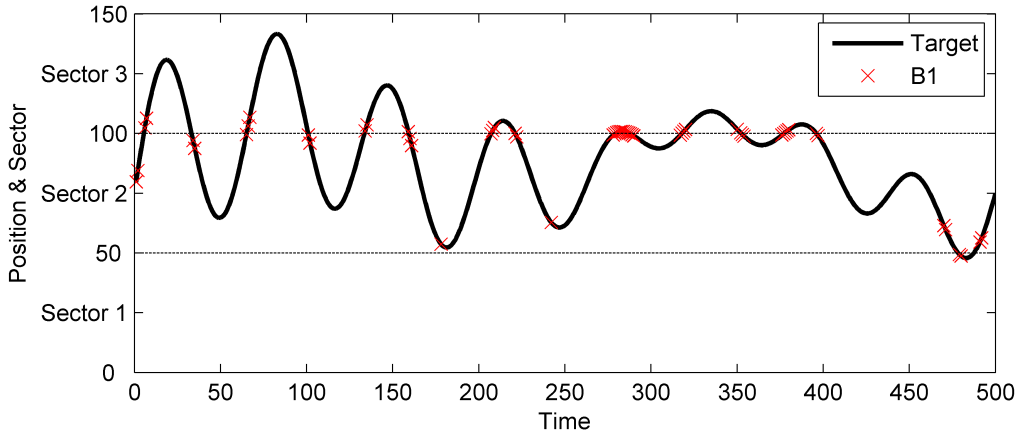


Figure 4: The line is the track of the target. Communication with this method causes process B_1 to communicates the position of the target as it gets close to a border.

For now, the focus has been on interaction between simulated components. Currently we are conducting real scale experiments on actual hardware within our laboratory. We expect to see that the interaction between different components as modeled in the JDL fusion model will benefit significantly and that system/sensor management will thereby be more effective correspondingly.

References

[1] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks (research note). *Artificial Intelligence*, 42(2-3):393–405, 1990.

[2] Veerle M. H. Coupé, Linda C. Van Der Gaag, and J. Dik F. Habbema. Sensitivity analysis: an aid for belief-network quantification. *Knowledge Engineering Review*, 15(3):215–232, 2000.

[3] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[4] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[5] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[6] Jan Willem Marck, Leon Kester, Jeroen Bergmans, Miranda van Iersel, and Eelke van Foeken. Effective and efficient communication of information. In *MultiSensor Fusion and Integration for Intelligent Systems*, 2008.

[7] L. Niklasson, M. Riveiro, F. Johansson, A. Dahlbom, G. Falkman, T. Ziemke, Ch. Brax, Th. Kronhamn, M. Smedberg, H. Warston, and P. Gustavsson. Revisions and extensions to the JDL data fusion model II. In *Proceedings of the Seventh International Conference on Information Fusion*, volume II, pages 1218–1230, Mountain View, CA, Jun 2004. International Society of Information Fusion.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Evolving Fixed-parameter Tractable Algorithms

Stefan A. van der Meer [a] Iris van Rooij [a,b] Ida Sprinkhuizen-Kuyper [a,b]

[a] *Radboud University Nijmegen, Department of Artificial Intelligence*
[b] *Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour*

Abstract

One effective means of computing NP-hard problems is provided by fixed-parameter tractable (fpt-) algorithms. An fpt-algorithm is an algorithm whose running time is polynomial in the input size and superpolynomial only as a function of an input parameter. Provided that the parameter is small enough, an fpt-algorithm runs fast even for large inputs. In this paper, we report on an investigation of the evolvability of fpt-algorithms via Genetic Programming (GP). The problem used in this investigation is the NP-hard 2D-Euclidean Traveling Salesman Problem (TSP), which is known to be fpt if the number of points *not* on the convex hull is taken as the parameter. The algorithm evolved in our GP study turns out to have clear characteristics of an fpt-algorithm. The results suggest GP can be utilized for generating fpt-algorithms for NP-hard problems in general, as well as for discovering input parameters that could be used to develop fpt-algorithms.

1 Introduction

Many computational problems, including those figuring in computational cognitive theories, are NP-hard. Traditionally, such NP-hard problems are considered intractable for all but small input sizes [3]. This has led applied computer scientists to focus attention on developing inexact (heuristic) methods for approaching NP-hard problems, and cognitive scientists to reject NP-hard problems as psychologically implausible models of human cognition [14]. However, it is known that certain NP-hard functions can be computable in a time that is polynomial for the overall input size and superpolynomial for only a small aspect of the input, called the *parameter*. Problems for which this holds are called *fixed-parameter tractable* and are said to belong to the complexity class FPT [2]. As long as the parameter is small enough for those instances of interest, an NP-hard problem in FPT can be considered efficiently solvable.

How do we know if a given problem is in FPT for some parameter k? One way of finding out is by designing an algorithm that solves the problem and establish that its running time can be expressed as a polynomial function of the input size, n, and a superpolynomial function of k (i.e., time $O(n^{\alpha} f(k))$, where α is some constant and $f(.)$ is a function depending only on k). Designing such an algorithm can be technically quite challenging, however, especially if the relevant parameter is yet to be discovered. It is for this reason that we investigate here the utility of genetic programming (GP) as a general method for developing or discovering fpt-algorithms for NP-hard problems.

Genetic programming (GP) is an evolutionary computation technique used to evolve computer programs [6, 13]. Populations of programs are evaluated and the fittest individuals are 'bred' to form new populations. Breeding is performed by applying genetic operations such as *crossover*, which creates new programs by recombining random parts of two selected programs, and *mutation*, where a random part of a program is randomly altered to form a new program. We used GP to evolve an algorithm that solves instances of the 2-dimensional Euclidean Traveling Salesman problem (TSP): given a set of points ('cities') in the plane, find the shortest tour visiting all points and returning to its starting point. This problem is known to be NP-hard and in FPT if the number of inner points is taken as a parameter [1]. The inner points of a TSP instance are the points that are in the interior of the convex hull. GP has often been applied to finding heuristic algorithms for TSP (see for example [12]), but to our knowledge no attempts to use GP to find fpt-algorithms exist to this date.

The aim of this research is to test whether or not an fpt-algorithm for TSP can be evolved using GP. Also of interest is whether GP can be used to discover potentially interesting input parameters for use in

developing new fpt-algorithms for NP-hard problems in general. The rest of this paper is organized as follows. Section 2 describes our method, Section 3 provides our results and Section 4 concludes.

2 Method

In GP, when evaluating an evolved program, called an *individual*, it is executed within a context of predefined supporting code, referred to as the *environment*. As this environment does not evolve, its functionality remains constant over all evaluations. The environment combined with the individual forms the algorithm. The *primitive set* is the set of functions, variables and constants available to the GP process for generating programs. As tree-based GP [13] was used, functions are referred to as function nodes (forming internal nodes in a program's tree), while variables and constants are terminal nodes (forming the leaves). Lastly, where the primitive set and environment define the search space, it is the *fitness function* that defines the goal of the search process [13], with individuals assigned a higher fitness value having a higher chance of being selected for breeding.

2.1 The environment

For the environment a structure was chosen similar to the one used in [12]. In this environment, a tour is built step by step, with the evolved individual forming a function that determines for each step what the next city in the tour should be. Algorithm 1 contains a pseudocode version of the environment. For each TSP instance used to evaluate a given individual, the environment loops through all cities in the problem instance that have not yet been 'visited' (i.e., that are not yet part of the tour). For each city, the evolved function calculates a score. When all unvisited cities have been scored, the city with the lowest score is selected. This city is added to the tour, and is therefore considered 'visited'. This process repeats itself until the tour includes all cities in the problem instance. In effect the algorithm 'travels' from city to city until it has visited them all and the tour is complete. If at each step the evolved function has given the best score to the correct city, the algorithm has found an optimal tour. In case of a Nearest Neighbor heuristic, it would score each city according to its distance from the 'current' city (i.e., the distance from the city last added to the tour to the city being evaluated), which will not yield an optimal tour for many problem instances.

Algorithm 1 The environment represented in pseudocode.

```
city-start = random city
city-current = city-start
while not visited all cities do
    selected = None
    bestscore = ∞
    for all unvisited cities do
        city-eval = next unvisited city to evaluate
        score = result of evolved function
        if score < bestscore then
            bestscore = score
            selected = city-eval
        end if
    end for
    add selected city to tour
    city-current = selected
end while
return length of tour
```

This structure was chosen because it constrains the evolved function to the specific task of solving TSP. The structure allows for a wide range of solvers, from purely heuristic (e.g., Nearest Neighbor) variations, to optimal exhaustive searchers. Which exact algorithms can be constructed is constrained by the primitive set.

Name	`distance`
Return type	Number
Child nodes	2 of type city
Description	Returns the distance between the two given cities.
Name	`if-on-convex`
Return type	Number
Child nodes	1 of type city, 2 of type number
Description	If the given city is on the convex hull, returns the result of evaluating the first numeric child node. Else it returns the result of the second.
Name	`for-loop-`X
Return type	Number
Child nodes	1 of type number
Description	Loops through unvisited cities, evaluating the child node for each one and adding up the result to a running total. This total is returned. X is a number referring to the associated variable-node, see Table 2.

Table 1: Function set, domain-specific functions

2.2 The primitive set

Tree-based GP was used, so an evolved algorithm forms a tree structure consisting of any valid combination of function nodes and terminal nodes. Strong typing was used to enforce type requirements of certain nodes [11]. At its root, the tree returns a real number: the calculated score. All function nodes and many terminal nodes return this type, meaning they can form the root of the tree, allowing for a wide variety of possible programs.

The basic concept behind the primitive set was also inspired by the research of [12], in that the primary tool for scoring a city to be used in the evolved function was *distance*. However, unlike their research, in our experiment we required the primitive set to be sufficient for specific types of algorithms other than heuristics such as Nearest Neighbor. The first type is exhaustive algorithms, the second is fpt-algorithms.

2.2.1 Iteration and recursion

Some form of iteration or recursion is needed in order for an evolved program to implement something more complex than a heuristic of a complexity that is linear to the size of its tree. Therefore, both iteration and recursion were implemented and added to the primitive set.

Traditional implementations of iteration in GP [8, 9] do not allow for nested loops, nor do they allow for a loop counter or element reference to be used by other nodes in the tree when iterating over a vector. In this research, complex nested loop structures are of interest, as they allow for more advanced calculations, and more emergent computational complexity. Therefore, a simplified version of the iteration implementation described in [4, 5] was used. A for-loop node was implemented which iterates over all unvisited cities when called. On each iteration, it sets a variable to reference the current unvisited city in the iteration, and evaluates its child node. The result is added to a running total, which is returned at the end of the loop. The variable can be accessed through a special terminal node that can only be generated inside the subtree of a given for-loop node, as it will be linked to that specific loop node and only has a value inside its 'scope'.

Recursion was implemented in the form of a terminal node. When this node is called while calculating the score for the city under evaluation, it adds the evaluated city to the tour and recursively calls the function holding the algorithm. By doing this, it causes the calculation of the rest of the tour as the algorithm would find it, if the given city were to be added to the tour. The length of this tour is returned, and the node returns this in turn as its result, after removing the evaluated city from the tour.

2.2.2 Primitive set contents

With the evolved program calculating a score using real numbers, it seemed useful to include basic mathematical operations for calculating and combining results from other nodes. The function set contained function nodes for addition, subtraction, multiplication and division, and for min and max operations. All these nodes require two children returning numbers, and return a single number themselves after performing their mathematical operation on the results of the child nodes. Besides these basic operations, certain

Name	`city-current`
Type	City
Description	The current city, i.e., the city last added to the tour.
Name	`city-eval`
Type	City
Description	The city currently being evaluated, i.e., the city that is being scored.
Name	`city-start`
Type	City
Description	The starting city of the tour (static).
Name	`var-`X
Type	City
Description	Associated with a for-loop node further up the tree (identifiable by identical X, see Table 1). Refers to the city that the loop has assigned to the variable before evaluating the subtree this node is in.
Name	`recursion`
Type	Number
Description	Returns the distance of the tour that the algorithm would travel if the city that is being evaluated were selected and traveled to by the algorithm.

Table 2: Terminal set

domain-specific functions were necessary, listed in Table 1. The `distance` node is used to find the distance between two cities. As arguments for this function, several terminal nodes exist that return a given city, listed in Table 2. The relevant nodes for iteration and recursion are also included, and are as defined earlier. Lastly, a node was added that represents the knowledge of the convex hull of a given TSP instance. This `if-on-convex` node checks if a given city is on the convex hull. If so, then it evaluates the first of its subtrees; if not, then it evaluates the other. Therefore, an evolved program using this node can alter its method of calculating a score (and its result) depending on whether the evaluated city is on the convex hull or not.

2.3 Fitness function

Traditionally, GP experiments use a single value to determine the fitness, such as the difference between the length of a shortest tour (i.e., the optimal solution for TSP) and the length of the tour that was found by an individual. In this experiment, however, there are two relevant values: speed and accuracy.

Speed was measured using the number of tree nodes evaluated in creating an individual's tour, where a lower number of evaluations is faster and therefore better (on this measure). Individuals with many loops or with a recursion would evaluate a larger number of nodes, and score worse than a Nearest Neighbor-like individual. Equation 1 shows how the speed measure was calculated, where the number of instances refers to the instances used in evaluating the individual.

$$fitness_{speed} = \frac{\text{number of instances}}{\text{nodes evaluated}} \tag{1}$$

Accuracy was measured as the difference between the length of an optimal solution to the TSP instance the individual just solved, and the length of the tour the individual found. Equation 2 shows the exact calculation.

$$fitness_{accuracy} = \frac{1}{1 + \text{ tour length error}} \tag{2}$$

Note that Equations 1 and 2 are chosen such that $fitness_{speed}$ is decreasing in the number of evaluated nodes, and $fitness_{accuracy}$ is decreasing in the length of the produced tour.

In exploratory runs it became clear that, if both $fitness_{speed}$ and $fitness_{accuracy}$ independently contributed to overall fitness, then the Nearest Neighbor (NN) individuals and exhaustive search individuals, consisting of only three and one nodes respectively, would always be selected for breeding. Apparently their good speed and good accuracy respectively would always 'beat' more complex individuals that were in their initial stages of development. This made it practically impossible for more complex individuals to exist for

longer than a single generation, and therefore difficult for such individuals to evolve into more 'fit' variants. To prevent the search process from fixating on the two extremes of exhaustive versus NN search, lower limits were set on both speed and accuracy. These limits would start at a high level in the beginning of the run, but would become lower with each generation until (i) the accuracy limit would make Nearest Neighbor search unfeasible and (ii) the speed limit would make exhaustive search unfeasible. It was our expectation that the introduction of such strict lower limits would enable the evolution process to go beyond the fastest heuristic approach and the intractable exact approach, and explore instead accurate yet tractable algorithms such as fpt-algorithms. The fitness functions with the additional lower limits are given in Equation 3.

$$\begin{gathered}\text{nodes evaluated} > \text{maximum nodes} \vee \\ \text{tour length error} > \text{maximum error} \Rightarrow \\ fitness_{speed} = fitness_{accuracy} = 0\end{gathered} \tag{3}$$

$$\text{otherwise} \Rightarrow fitness_{speed} = \frac{\text{number of instances}}{\text{nodes evaluated}} \wedge fitness_{accuracy} = \frac{1}{1 + \text{ tour length error}}$$

In comparing two individuals, it is very likely that neither of them may be better in both speed and accuracy, particularly in the earlier generations of a run, and especially considering the existence of the extreme individuals mentioned earlier. A simple criterion was introduced to counteract this effect: Whenever individuals were compared during selection, if neither was better on both speed and accuracy, there would be a chance they were then compared on *either* speed *or* accuracy to find a winner. This probability was set to start at a high level, and decreased as the process advanced in generations.

2.4 Experiment details

The GP experiment was implemented using the evolutionary computation for Java toolkit, ECJ [10]. Many GP parameters were left at the defaults used by Koza [7], such as those involving the building of initial trees in the population. Experiment-specific parameters and their values are listed in Table 3.

Parameter	**Value**
Generations	50
Population size	128
Crossover rate[1]	0.80
Mutation rate[1]	0.10
Reproduction rate[1]	0.10
TSP instance size	7
TSP instances per evaluation	50
Total pool of random instances	500

[1]The crossover, mutation and reproduction rates determine the probability of said genetic operation being used in breeding a new individual. See [7, 13] for details on these operations.

Table 3: GP experiment parameters

The TSP instances used in the experiment each consisted of 7 points. This number was kept deliberately low to ensure that evaluation progressed at a reasonable rate. Larger instances meant that individuals using (exhaustive) recursion would spend a large amount of time per instance, making the evaluation of a large number of individuals take an impractical amount of time. Population size was limited to 128 for the same reason. The instances were generated beforehand, as a set of 7 random coordinates in an area of 500 by 500 points. Every possible number of inner points (0 to 4, as a minimum of 3 points form the convex hull) was equally represented in the pool of instances. This was achieved by randomly generating point sets and disregarding instances that did not match the required number of inner points. For each generated instance both the optimal solution and the points on the convex hull were calculated. Each individual was evaluated on 50 TSP instances randomly selected from a pool of 500 available instances.

3 Results

3.1 Best evolved individual

In the GP experiment, one type of individual was consistently selected as the best individual of a run. The individual's code is shown in Program 1. This individual would generally develop fairly early in the run, between generations 10 and 25 (of 50), and due to its relatively high fitness it would immediately form the best individual of the generation and remain so until the end of the run. The early development is not surprising given the structure of the individual. It is a small tree consisting of only a few nodes, and substantial parts of the tree are formed by nodes that make up two common individuals in the population. The subtree `(distance city-current city-eval)` is equal to the tree of a Nearest Neighbor (NN) individual, shown in Program 2. Similarly, the `(recursion)` node would on its own form the entirety of an exhaustive search individual, shown in Program 3.

Program 1 The program of the best evolved individual.

```
(if-on-convex city-current
    (distance city-current city-eval)
    (recursion))
```

Program 2 The program of the common Nearest Neighbor individual.

```
(distance city-current city-eval)
```

Program 3 The program of the common exhaustive search individual.

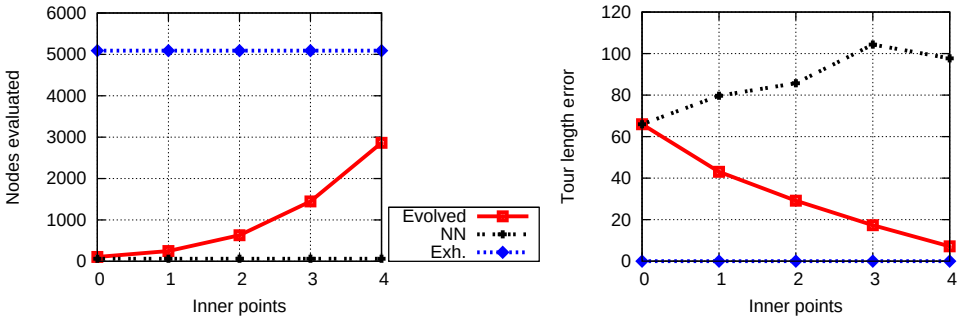
```
(recursion)
```

The behavior of the best evolved individual (Program 1) is straightforward: If the current city is part of the convex hull, it travels to the nearest neighboring city. Otherwise, it recursively builds possible extensions of the tour resulting from travelling to any of the unvisited cities. This recursive process repeats *until* the program encounters again a point on the convex hull, in which case it will extend each of the partial tours constructed so far by traveling to the nearest unvisited neighbor. From all tours constructed in the process, the program determines which is the shortest, and travels to the unvisited city that has that tour as a result. Note that if the instance the algorithm is solving happens to have few inner points, say only 1, it will do much less recursion than an exhaustive solver. At the same time, it will give more accurate results than NN when solving more complex instances with multiple inner points, as the 'look ahead' in the recursion allows it to avoid certain bad choices that NN would make. If tours with such bad choices occur after a recursion has been entered, they will most likely be discarded due to their higher length.

3.2 Fpt-characteristics

Is the best evolved individual an fpt-algorithm for TSP? To address this question, we first consider the algorithm's time behavior: The worst-case[2] time-complexity of the best individual is $O(k!\,(n-k)^2) = O(k!\,n^2)$, where n is the total number of points and k is the number of inner points. For instances with zero inner points, the program behaves as an NN individual (with time-complexity $O(n^2)$), and for instances with a very large number of inner points, performance is nearer an exhaustive search (with time-complexity $O(n!)$). We also investigated the algorithm's average-case time-complexity by running the algorithm on the 500 random instances in the pool. The results, depicted in Figure 1a, show that the average time required for Program 1 to find a tour grows speedily with the number of inner points, with its running time being close to that of the NN heuristic for few inner points and growing closer to the exhaustive algorithm as the number of inner points increase. In sum, the evolved program indeed exploits the number of inner points for

[2]Due to the nature of both the environment and the evolved program itself, an individual's time-complexity depends on the point selected as starting point (which is a random selection). If this point is an inner point, for example, more recursion will be performed than if that city is not visited until later on in the computation.

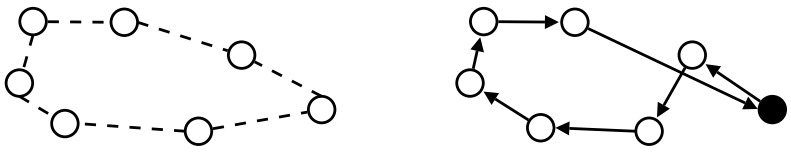


(a) Average number of nodes evaluated (b) Average tour length error

Figure 1: Speed (a) and accuracy (b) as a function of number of inner points, for the evolved program, NN, and exhaustive search, averaged over all 500 random instances and all possible starting cities.

the efficient computation of instances for which that parameter is small, and the running-time behavior is as one expects of an fpt-algorithm (i.e., the running time can be expressed as a polynomial function of input size, n, and a superpolynomial function of only the parameter k).

As it turns out, the best evolved individual does not meet the second criterion for being an fpt-algorithm, viz., exactness. Due to its reliance on NN to select the optimal tour when travelling over the convex hull, it inherits some of NN's flaws. Figure 2 shows an example of a trivial instance (i.e., one without any inner points) where, for a certain starting point, NN fails to find an optimal tour. Such instances are not rare: For only 4 of the 100 generated instances with no inner points, NN is able to find an optimal tour regardless of the starting point, with on average 3.32 out of 7 starting points per instance resulting in a less than optimal tour. Be that as it may, the performance of the best evolved individual is much better than NN for all instances with at least one inner point (see Figure 1b). Hence, even if the best evolved individual is not an exact algorithm for TSP, it clearly outperforms a polynomial-time heuristic, like NN.



(a) Optimal tour (convex hull) (b) NN tour

Figure 2: Example instance with no inner points where NN does not find an optimal tour when starting from a certain city (starting city shown in black).

4 Conclusion

The program evolved in our GP experiment shows clear characteristics of an fpt-algorithm, even though strictly speaking it is not: The program does not solve all TSP instances optimally, though it is much more accurate than its polynomial-time competitor, the NN heuristic. Also, the program is characterized by an fpt running time of $O(k!\, n^2)$. This result is promising with regards to the utility of GP for developing fpt-algorithms for NP-hard problems in general and discovering relevant parameters that can be used in such algorithms. We think that the fact that the best individual in our experiment was not an exact algorithm for TSP does not detract from this point, because an evolved fpt-heuristic can give a clear suggestion as to the direction in which an fpt-algorithm can be sought. After all, TSP is known to be in FPT if the parameter is the number of inner points [1], and the fpt-heuristic evolved in our experiment used this same parameter to bound its superpolynomial running time. Besides the important step of discovering the parameter, it is conceivable that fpt-like inexact individuals themselves can be transformed into fpt-algorithms; and even if they cannot, an evolved (inexact) fpt-heuristic may still strike a better balance between speed and accuracy for instances of practical interest, than do available polynomial-time heuristics.

References

[1] V.G. Deineko, M. Hoffman, Y. Okamoto, and G.J. Woeginger. The traveling salesman problem with few inner points. In K.-Y. Chwa and J.O. Munro, editors, *Computing and Combinatorics: 10th Annual International Conference, COCOON 2004*, volume 3106 of *LNCS*, pages 268–277, Berlin, 2004. Springer-Verlag.

[2] M.R. Fellows and R.G. Downey. Parameterized complexity after (almost) 10 years: review and open questions. In C.S. Calude et al., editors, *Proceedings of Combinatorics, Computation and Logic, DMTCS'99 and CATS'99*, volume 21 of *Australian Computer Science Communications*, pages 1–33, Singapore, 1999. Springer-Verlag.

[3] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, New York, NY, 1979.

[4] E. Kirshenbaum. Genetic programming with statically scoped local variables. In D. Whitley et al., editors, *GECCO 2000: Proceedings of the Genetic and Evolutionary Computation Conference, July 10-12, 2000, Las Vegas, Nevada*, pages 459–468, San Francisco, CA, 2000. Morgan Kaufman.

[5] E. Kirshenbaum. Iteration over vectors in genetic programming. Technical Report HPL-2001-327, HP Laboratories Palo Alto, 2001.

[6] J.R. Koza. Genetic Programming: a paradigm for genetically breeding populations of computer programs to solve problems. Technical Report CS-TR-90-1314, Stanford University, Stanford, CA, 1990.

[7] J.R. Koza. *Genetic Programming II: automatic discovery of reusable programs*. MIT Press, Cambridge, MA, 1994.

[8] J.R. Koza and D. Andre. Evolution of iteration in genetic programming. In L.J. Fogel, P.J. Angeline, and T. Baeck, editors, *Evolutionary Programming V: proceedings of the fifth annual conference on evolutionary programming*, Cambridge, MA, 1996. MIT Press.

[9] J.R. Koza, D. Andre, F.H. Bennett III, and M. Keane. *Genetic Programming 3: Darwinian invention and problem solving*. Morgan Kaufman, 1999.

[10] S. Luke et al. *ECJ 17 - A Java-based evolutionary computation system*, 2008. Retrieved from http://www.cs.gmu.edu/ eclab/projects/ecj/.

[11] D.J. Montana. Strongly typed genetic programming. *Evolutionary Computation*, 3(2):199–230, 1995.

[12] M. Oltean and D. Dumitrescu. Evolving TSP heuristics using multi expression programming. In M. Bubak et al., editors, *Computational Science - ICCS 2004: 4th International Conference, Proceedings, Part II*, volume 3037 of *LNCS*, pages 670–673, Berlin, 2004. Springer-Verlag.

[13] R. Poli, W.B. Langdon, N.F. McPhee, and J.R. Koza. Genetic Programming: an introductory tutorial and a survey of techniques and applications. Technical Report CES-475, University of Essex, Computing and Electronic Systems, United Kingdom, 2007.

[14] I. van Rooij. The tractable cognition thesis. *Cognitive Science*, 32:939–984, 2008.

Lambek–Grishin Calculus Extended to Connectives of Arbitrary Arity

Matthijs Melissen [a]

[a] *Universiteit Utrecht, P.O.Box 80126 3508 TC Utrecht*

Abstract

This paper introduces Lambek–Grishin calculus for n-ary connectives, which can be seen as a generalization of binary and unary Lambek–Grishin calculus. It is a categorial grammar that is at least mildly context-sensitive, and allows for branching of arbitrary arity. Therefore it seems very suitable for linguistic applications. A cut-free presentation of the calculus is showed, proving the decidability of the calculus. Finally we investigate the symmetries of the calculus by making use of group theory.

1 Introduction

One of the disadvantages of traditional categorial grammars like standard Lambek calculus [8] and its symmetric extension, the binary Lambek–Grishin calculus [9], is that they only allow for binary branching. However, in many linguistic examples it seems to be more natural to also admit ternary branching or branching of even higher order. Consider for example sentences with ditransitive verbs like ′He gives the man a book′. We might want to see the phrase ′gives the man a book′ as a branch with three direct constituents, namely ′gives′, ′the man′ and ′a book′.

One of the solutions to this problem is the extension of Lambek calculus to an n-ary variant. For example Buszkowski [2] proposed a system with residuation modalities of arbitrary arity, but in [6] it was proved that this calculus recognizes only the context-free languages. Therefore this formalism might not be a good choice for natural language, as it has been argued that natural language is not context-free [5].

In this paper, we present a categorial grammar that has both advantages: it is (at least) mildly context-sensitive, and still we have the possibility of describing branching or arbitrary arity. The calculus can be seen as a generalization of the binary Lambek–Grishin calculus [9] and the unary Lambek–Grishin calculus [3] to arbitrary arity. Binary Lambek–Grishin calculus is mildly context-sensitive (and parsable in polynomial time) [11].

In the next section, I will present the Lambek–Grishin calculus for unary and binary connectives. In section 3, I introduce the Lambek–Grishin calculus for n-ary arity. In section 4, I show that this calculus is decidable. In section 5 I study the symmetries of the calculus with aid of group theory. Finally I draw some conclusions in section 6, and point out some lines for further research.

2 Lambek–Grishin calculus for binary and unary connectives

2.1 The binary Lambek–Grishin calculus

Definition 1 (Minimal Lambek–Grishin calculus)**.** The *Lambek–Grishin calculus* is an extension of the *nonassociative Lambek calculus* [8]. Lambek–Grishin calculus is, just like the ordinary Lambek calculus, a deductive system. A sentence is grammatical if and only if the sentence can be derived in the deductive system. To every word in the language a finite number of types is assigned. We have a set of basic types called *Atoms*. The set of *types* T is defined as follows: every atom is a type, and if a and b are types, then the following formulas are types as well:

$$a \otimes b, \quad a \backslash b, \quad a / b, \quad a \oplus b, \quad a \oslash b, \quad a \odot b.$$

Intuitively we can see the $\otimes$-operator as concatenation of words. Furthermore one or more words with type $a\backslash b$ can be seen as a group of words which need a phrase of type a on their left, and return a phrase of type b. In the same way we can see a phrase of type a/b as a phrase which asks for a phrase of type b on its right-hand side, before returning an a type phrase. Furthermore we can see $\oplus$, $\oslash$ and $\odot$ as the duals of $\otimes$, $/$ and $\backslash$ under arrow reversal. We define the *size* of a type a, denoted $|a|$, as the number of connectives it contains.

Minimal Lambek–Grishin calculus [9] for binary connectives $\mathbf{LG_2^\emptyset}$ is defined as the calculus having the following rules:

$$a \to a \qquad \frac{a \to b \qquad b \to c}{a \to c}\ \text{Cut} \qquad \frac{\dfrac{b \to a\backslash c}{a \otimes b \to c}\ \text{Res}}{a \to c/b}\ \text{Res} \qquad \frac{\dfrac{a \oslash c \to b}{c \to a \oplus b}\ \text{Res}}{c \odot b \to a}\ \text{Res}$$

From left to right: *identity*, *transitivity* (also named '*cut*'), and *residuation* (2x). With the double line we indicate derivability in both directions. In the cut rule, we call b the *cut formula*. The *complexity* of a cut is defined as $|a| + |b| + |c|$.

Definition 2 (Lambek–Grishin calculus with interaction)**.** Grishin [4] proposes four classes of postulates to govern the interaction between the $\oplus$ and $\otimes$ connectives. The Lambek–Grishin calculus $\mathbf{LG}$, also notated as $\mathbf{LG_2^\emptyset + IV}$, is the extension of the minimal calculus $\mathbf{LG_2^\emptyset}$ with the postulates from Grishin class IV added.

$$\frac{a \oslash (b \otimes c) \to d}{(a \oslash b) \otimes c \to d}\ G1 \qquad \frac{b \oslash (a \otimes c) \to d}{a \otimes (b \oslash c) \to d}\ G2 \qquad \frac{(a \otimes b) \odot c \to d}{a \otimes (b \odot c) \to d}\ G3 \qquad \frac{(a \otimes c) \odot b \to d}{(a \odot b) \otimes c \to d}\ G4$$

Definition 3 (Lambek grammar)**.** A *Lambek grammar* has the form $\mathcal{L}(\Sigma, s, f)$, such that for all $t \in \Sigma$, $f(t) \subset T$ and $f(t)$ is finite, where T is the set of all types of the Lambek calculus. We define the *yield* of a formula consisting of only $\otimes$-connectives as follows: $\mathrm{yield}(a) = a$ if a is an atom, $\mathrm{yield}(a \otimes b) = \mathrm{yield}(a)\mathrm{yield}(b)$. The *language generated by the Lambek grammar* $\mathcal{L}(\Sigma, s, f)$ is defined as the set of all expressions $t_1 \dots t_n$ over the alphabet Σ for which there exists a derivable sequent with yield $b_1 \dots b_n \to s$ such that $b_i \in f(t_i)$ for all $i \leq n$.

2.2 Symmetries

Definition 4 (Order-preserving symmetry)**.** We define on formulas the operations $\bowtie$, $\sharp$ and $\flat$. The operation $\bowtie$ exchanges the order of all types, while $\sharp$ and $\flat$ only exchange the order of the $\otimes$- and $\oplus$-family connectives, respectively. Formally the operations are defined as follows:

$$\begin{array}{rcl} a^{\bowtie} &=& a \quad \text{if } a \in Atoms \\ (a \otimes b)^{\bowtie} &=& b^{\bowtie} \otimes a^{\bowtie} \\ (a/b)^{\bowtie} &=& b^{\bowtie}\backslash a^{\bowtie} \\ (a\backslash b)^{\bowtie} &=& b^{\bowtie}/a^{\bowtie} \\ (a \oplus b)^{\bowtie} &=& b^{\bowtie} \oplus a^{\bowtie} \\ (a \oslash b)^{\bowtie} &=& b^{\bowtie} \odot a^{\bowtie} \\ (a \odot b) &=& b^{\bowtie} \oslash a^{\bowtie} \end{array} \qquad \begin{array}{rcl} a^{\sharp} &=& a \quad \text{if } a \in Atoms \\ (a \otimes b)^{\sharp} &=& b^{\sharp} \otimes a^{\sharp} \\ (a/b)^{\sharp} &=& b^{\sharp}\backslash a^{\sharp} \\ (a\backslash b)^{\sharp} &=& b^{\sharp}/a^{\sharp} \\ (a \oplus b)^{\sharp} &=& a^{\sharp} \oplus b^{\sharp} \\ (a \oslash b)^{\sharp} &=& a^{\sharp} \oslash (b) \\ (a \odot b)^{\sharp} &=& a^{\sharp} \odot b^{\sharp} \end{array} \qquad \begin{array}{rcl} a^{\flat} &=& a \quad \text{if } a \in Atoms \\ (a \otimes b)^{\flat} &=& a^{\flat} \otimes b^{\flat} \\ (a/b)^{\flat} &=& a^{\flat}/b^{\flat} \\ (a\backslash b)^{\flat} &=& a^{\flat}\backslash b^{\flat} \\ (a \oplus b)^{\flat} &=& b^{\flat} \oplus a^{\flat} \\ (a \oslash b)^{\flat} &=& b^{\flat} \odot (a) \\ (a \odot b)^{\flat} &=& b^{\flat} \oslash a^{\flat} \end{array}$$

Theorem 5 ([9])**.** *It holds that $a^{\bowtie} \to b^{\bowtie}$ if and only if $a \to b$.*

Theorem 6 ([9])**.** *The operations $\bowtie$, $\sharp$ and $\flat$ form together with the identity transformation a group under function composition isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.*

Definition 7 (Order-reversing symmetry)**.** Another operator is ∞, with the following definition. We have $a^\infty = a$ whenever a is an atom. Furthermore we define

$$\begin{array}{rclrcl} (a \otimes b)^\infty & = & b^\infty \oplus a^\infty; & (a \oplus b)^\infty & = & b^\infty \otimes a^\infty; \\ (a/b)^\infty & = & b^\infty \oslash a^\infty; & (a \oslash b)^\infty & = & b^\infty \backslash a^\infty; \\ (a\backslash b)^\infty & = & b^\infty \oslash a^\infty; & (a \oslash b)^\infty & = & b^\infty / a^\infty. \end{array}$$

Theorem 8 ([9]). *It holds that $a^\infty \to b^\infty$ if and only if $b \to a$.*

Theorem 9 ([9]). *The group generated by ∞, $\flat$ and $\sharp$ is isomorphic to D_4, the dihedral group with 8 elements.*

2.3 The unary Lambek–Grishin calculus

Definition 10. Lambek–Grishin calculus with unary connectives was proposed in [3]. The minimal unary Lambek–Grishin calculus $\mathbf{LG}_\mathbf{1}^\emptyset$ contains the following types: every atom is a type, and if a is a type, then $\Diamond_1 a$, $\Box_1 a$, $\Diamond_2 a$ and $\Box_2 a$ are types as well. Again we have the axiom $a \to a$ and transitivity. The residuation rules for the unary calculus are defined in the following way:

$$\forall i \in \{1, 2\} \quad \Diamond_i a \to b \text{ iff } a \to \Box_i b.$$

Just like for the binary variant, we can add interaction postulates, which gives us $\mathbf{LG}_\mathbf{1}^\emptyset + \mathbf{IV}$. In the unary case, we have only one interaction postulate (appearing in three interderivable forms, though). Therefore we have only the trivial group with one element. One way of writing down this interaction postulate is the following:

$$\Diamond_1 \Diamond_2 a \to \Diamond_2 \Diamond_1 a.$$

3 Lambek–Grishin calculus for n-ary connectives

Now we have seen the binary and unary Lambek–Grishin calculus, we can extend the Lambek–Grishin calculus to arbitrary arity.

Definition 11. We define the types for n-ary Lambek–Grishin calculus $\mathbf{LG}_\mathbf{n}^\emptyset$ as follows. Again we define a set of atoms. Every atom is a type, and if a_k is a type for every k such that $1 \leq k \leq n$ then $f_\bullet(a_1, \ldots, a_n)$, $f^i_\to(a_1, \ldots, a_n)$, $g_\bullet(a_1, \ldots, a_n)$ and $g^i_\to(a_1, \ldots, a_n)$ are types for every i such that $1 \leq i \leq n$. We should see f and g as galois connected pairs, i.e. $f_\bullet(a_0, \ldots, a_n) \to b$ if and only if $b \to g_\bullet(a_0, \ldots, a_n)$, just like the relation between $\otimes$ and $\oplus$ in the binary calculus. Furthermore we should see $f_\to$ and $g_\to$ as the residue operators of $f_\bullet$ and $g_\bullet$, respectively. From a (linear) logical perspective, we can see the $\bullet$-connectives $f_\bullet$ and $g_\bullet$ as multiplication, while we can see the $\to$-connectives $f_\to$ and $g_\to$ as implication. The superscript i in the implications denotes which element is the goal type of the implication. We can see it as succedent or conclusion of the implication, while the other $n - 1$ elements can be seen as antecedents.

Remark 12. Of course, the connectives of binary and unary Lambek–Grishin calculus can be expressed in this notation as well. For binary Lambek–Grishin calculus we have the following correspondences:

$a \otimes b$	a/b	$a\backslash b$	$a \oplus b$	$a \oslash b$	$a \oslash b$
$f_\bullet(a, b)$	$f^1_\to(a, b)$	$f^2_\to(a, b)$	$g_\bullet(a, b)$	$g^1_\to(a, b)$	$f^2_\to(a, b)$

For unary Lambek–Grishin calculus, the correspondences are as follows:

$\Diamond_1$	$\Box_1$	$\Box_2$	$\Diamond_2$
$f_\bullet(a)$	$f^1_\to(a)$	$g_\bullet(a)$	$g^1_\to(a)$

The minimal Lambek–Grishin calculus for n-ary connectives $\mathbf{LG}_\mathbf{n}^\emptyset$ has the following axioms and rules:

$$a \to a \quad \text{(identity)} \tag{1}$$

$$\frac{a \to b \qquad b \to c}{a \to c}\ \text{Cut} \tag{2}$$

Residuation rules:

$$\frac{f_\bullet(a_1,\dots,a_n)\to b}{a_i\to f^i_{\to}(a_1,\dots,a_{i-1},b,a_{i+1},\dots,a_n)}\,\text{Res} \tag{3}$$

$$\frac{g^i_{\to}(a_1,\dots,a_{i-1},b,a_{i+1},\dots,a_n)\to a_i}{b\to g_\bullet(a_1,\dots,a_n)}\,\text{Res} \tag{4}$$

One can check easily that the rules for binary and unary Lambek–Grishin calculus are instances of these rules.

Definition 13. The *grammar* belonging to $\mathbf{LG}^{\emptyset}_{\mathbf{n}}+\mathbf{IV}$ can be defined in the same way as the grammar for $\mathbf{LG}^{\emptyset}_{\mathbf{2}}+\mathbf{IV}$ as defined in definition 3. However, the definition of yield will be adapted, such that the *yield* of a formula consisting of only $f_\bullet$ connectives is defined as follows: $\operatorname{yield}(a)=a$ if a is an atom, $\operatorname{yield}(f_\bullet(a_0,\dots,a_n))=\operatorname{yield}(a_0)\dots\operatorname{yield}(a_n)$.

3.1 Order-preserving symmetry

Now we will study the symmetries of $\mathbf{LG}^{\emptyset}_{\mathbf{n}}+\mathbf{IV}$. To do this, we define for all permutations π_f,π_g on $(1,\dots,n)$ a *permutation on formulas* $Z_{\pi_f\pi_g}$:

$$\begin{aligned}
Z_{\pi_f\pi_g}(a) &= a \quad \text{if } a \text{ is an atom;}\\
Z_{\pi_f\pi_g}(f_\bullet(a_1,\dots,a_n)) &= f_\bullet(Z_{\pi_f\pi_g}(a_{\pi_f(1)}),\dots,Z_{\pi_f\pi_g}(a_{\pi_f(n)}));\\
Z_{\pi_f\pi_g}(g_\bullet(a_1,\dots,a_n)) &= g_\bullet(Z_{\pi_f\pi_g}(a_{\pi_g(1)}),\dots,Z_{\pi_f\pi_g}(a_{\pi_g(n)}));\\
Z_{\pi_f\pi_g}(f^k_{\to}(a_1,\dots,a_n)) &= g^{\pi_f(k)}_{\to}(Z_{\pi_f\pi_g}(a_{\pi_f(1)}),\dots,Z_{\pi_f\pi_g}(a_{\pi_f(n)}));\\
Z_{\pi_f\pi_g}(g^k_{\to}(a_1,\dots,a_n)) &= g^{\pi_g(k)}_{\to}(Z_{\pi_f\pi_g}(a_{\pi_g(1)}),\dots,Z_{\pi_f\pi_g}(a_{\pi_g(n)})).
\end{aligned}$$

That is, we apply the π_f permutation to the f-family, and the π_g permutation to the g-family of connectives.

Theorem 14. *It holds that $Z_{\pi_f\pi_g}(a)\to Z_{\pi_f\pi_g}(b)$ if and only if $a\to b$ for all permutations π_f,π_g on $1,\dots,n$.*

Proof sketch. All axioms and rules of $\mathbf{LG}^{\emptyset}_{\mathbf{n}}$ are closed under permutation on formulas. □

Now we define a rotation operator $\bowtie(i,j)$ as Z_{π_f,π_g}, where $\pi_f(k)=k+i \mod n$ and $\pi_g(k)=k+j \mod n$. This operator rotates the arguments of connectives of the f-family i places, and of the g-family j places. This definition is an extension of the operators $\bowtie$, $\flat$ and $\sharp$ for binary Lambek–Grishin calculus in the following way, for $n=2$: $\bowtie=\bowtie(1,1)$, $\sharp=\bowtie(1,0)$ and $\flat=\bowtie(0,1)$.

3.2 Order-reversing symmetry

We can extend the symmetry function ∞ to our n-ary system:

$$\begin{aligned}
a^\infty &= a \quad \text{if } a \text{ is an atom;}\\
f_\bullet(a_1,\dots,a_n)^\infty &= g_\bullet(a_n^\infty,\dots,a_1^\infty);\\
g_\bullet(a_1,\dots,a_n)^\infty &= f_\bullet(a_n^\infty,\dots,a_1^\infty);\\
f^i_{\to}(a_1,\dots,a_n)^\infty &= g^{n-i}_{\to}(a_n^\infty,\dots,a_1^\infty);\\
g^i_{\to}(a_1,\dots,a_n)^\infty &= f^{n-i}_{\to}(a_n^\infty,\dots,a_1^\infty).
\end{aligned}$$

Theorem 15. *It holds that $a^\infty\to b^\infty$ if and only if $b\to a$.*

3.3 Interaction principles

We proceed by giving postulates which govern interaction between the f- and g-families. An example of such a postulate is

$$f_\bullet(a_1,\ldots,a_{n-1},g^1_\rightarrow(b_1,\ldots,b_n)) \to g^1_\rightarrow(f_\bullet(a_1,\ldots,a_{n-1},b_1),b_2,\ldots,b_n).$$

We obtain the full class of postulates by taking the closure of the given postulate under $\bowtie(i,j)$ such that $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$. This gives us the following postulates, parameterized by i and j:

$$\begin{array}{rl} & f_\bullet(a_1,\ldots,a_{i-1},g^j_\rightarrow(b_1,\ldots,b_n),a_{i+1},\ldots,a_n) \\ \to & g^j_\rightarrow(b_1,\ldots,b_{j-1},f_\bullet(a_1,\ldots,a_{i-1},b_j,a_{i+1},\ldots,a_n),b_{j+1},\ldots,b_n). \end{array} \tag{5}$$

Now we can see that this class contains exactly n^2 different postulates (those who are familiar with Grishin's original paper may note that we only considered Grishin class IV). The interaction principles defined in this way are generalizations of the interaction principles for both binary and unary Lambek–Grishin calculus. This explains also why we have four non-equivalent interaction principles per class in binary Lambek–Grishin calculus, while we have only one in unary Lambek–Grishin calculus.

4 Decidability of $\mathbf{LG^{\emptyset}_n + IV}$

Theorem 16. *Lambek–Grishin calculus for arbitrary modalities* $\mathbf{LG^{\emptyset}_n + IV}$ *is decidable.*

Proof. The decidability proof for n-ary Lambek–Grishin calculus is an extension to the decidability proof for binary Lambek–Grishin calculus [9], which in turn is an extension to the decidability proof for Lambek calculus [10]. Note however that we simplified these proofs somewhat, because we do not need the principal cases and the special cases for Grishin interactions.

The strategy for this proof is as follows. We introduce a new axiomatization for $\mathbf{LG^{\emptyset}_n + IV}$, which we prove to be equivalent to out original axiomatization. Next, we prove that in the new axiomatization, the transitivity rule can be eliminated.

Now we will describe the new axiomatization. We keep identity (1), transitivity (2) and the residuation rules (3, 4). However we replace the Grishin interactions by rule-based versions (6), and we add monotonicity (7, 8, 9, 10). The rule-based Grishin interactions look as follows:

For all $1 \leq i \leq n$, $1 \leq j \leq n$ we have:

$$\frac{g^i_\rightarrow(a_1,\ldots,a_{i-1},f_\bullet(b_1,\ldots,b_n),a_{i+1},\ldots,a_n) \to d}{f_\bullet(b_1,\ldots,b_{j-1},g^i_\rightarrow(a_1,\ldots,a_{i-1},b_j,a_{i+1},\ldots,a_n),b_{j+1},\ldots,b_n) \to d}\ \text{Gr} \tag{6}$$

We add the following monotonicity rules:

$$\frac{a_i \to b_i \text{ for all } i}{f_\bullet(a_1,\ldots,a_n) \to f_\bullet(b_1,\ldots,b_n)}\ \text{Mon} \tag{7}$$

$$\frac{a_k \to b_k \text{ for } k \neq i \quad b_i \to a_i}{f^i_\rightarrow(b_1,\ldots,b_n) \to f^i_\rightarrow(a_1,\ldots,a_n)}\ \text{Mon} \tag{8}$$

$$\frac{a_i \to b_i \text{ for all } i}{g_\bullet(a_1,\ldots,a_n) \to g_\bullet(b_1,\ldots,b_n)}\ \text{Mon} \tag{9}$$

$$\frac{a_k \to b_k \text{ for } k \neq i \quad b_i \to a_i}{g^i_\rightarrow(b_1,\ldots,b_n) \to g^i_\rightarrow(a_1,\ldots,a_n)}\ \text{Mon} \tag{10}$$

The old system, consisting of identity, transitivity, residuation, Grishin-interactions is equivalent to the new system, consisting of identity, transitivity, residuation, monotonicity and rule-based Grishin interactions. Grishin interactions and rule-based Grishin interactions can simply be derived from each other with use of transitivity. Now we show how to derive monotonicity. First note that from $a_i \to b_i$ for all i and $f_\bullet(b_1,\ldots,b_n) \to c$ we can derive $f_\bullet(a_1,\ldots,a_n) \to c$, by iteratively applying the following:

$$\dfrac{a_i \to b_i \qquad \dfrac{f_\bullet(a_1,\dots,a_{i-1},b_i,b_{i+1},\dots,b_n) \to c}{b_i \to f^i_\to(a_1,\dots,a_{i-1},c,a_{i+1},\dots,a_n)}\,\text{Res}}{\dfrac{a_i \to f^i_\to(a_1,\dots,a_{i-1},c,a_{i+1},\dots,a_n)}{f_\bullet(a_1,\dots,a_{i-1},a_i,b_{i+1}\dots,b_n) \to c}\,\text{Res}}\,\text{Cut}$$

We call this procedure $\mathbf{P}_\bullet$. In a similar way we can derive from $b_k \to a_k$ for $k \neq i$, $a_i \to b_i$ and $c \to f^i_\to(a_1,\dots,a_n)$ the formula $c \to f^i_\to(b_1,\dots,b_n)$. This procedure will be called $\mathbf{P}_\to$. From $\mathbf{P}_\bullet$ and $\mathbf{P}_\to$, we can easily obtain monotonicity.

Now we will show that we can eliminate transitivity (or cut) from the calculus consisting of identity (1), transitivity (2), residuation (3, 4), monotonicity (7, 8, 9, 10) and rule-based Grishin interactions 6. The resulting calculus has no rule with more (instances of) atoms in the hypotheses than in the conclusion, so it is decidable.

First we label every subformula with a polarity. We label $a \to b$ as $a^- \to b^+$. Furthermore, in positive context we have:

$$\begin{array}{ll} f_\bullet(a_1^+,\dots,a_n^+); & f^i_\to(a_1^-,\dots,a_{i-1}^-,a_i^+,a_{i+1}^-,\dots,a_n^-);\\ g_\bullet(a_1^+,\dots,a_n^+); & g^i_\to(a_1^-,\dots,a_{i-1}^-,a_i^+,a_{i+1}^-,\dots,a_n^-). \end{array}$$

In negative context all polarities are reversed. Note that residuation and the Grishin rules do not change the polarity of any subformula.

We show that the highest cut in the proof always can be eliminated, or replaced by n cuts with lower complexity, measured as $|a| + |b| + |c|$. We show this by case analysis on the cut formula of the uppermost cut.

Case 1: The cut formula of the uppermost cut is an atom. In this case, the antecedent of the right premise must be introduced by identity somewhere higher in the proof. Therefore the proof has the form on the left-hand side. As we can replace atoms by any formula, we can replace atom b by formula a, and rewrite the proof as follows:

$$\dfrac{a \to b \qquad \dfrac{\dfrac{b \to b}{\vdots}}{b \to c}}{a \to c} \quad\rightsquigarrow\quad \dfrac{\dfrac{a \to b}{\vdots}}{a \to c}$$

This eliminates the uppermost cut.

Case 2a: The cut formula of the uppermost cut has the form $f_\bullet(a_1,\dots,a_n)$. As the $f_\bullet$-connective can only be introduced in a positive context by monotonicity, the succedent of the left premise must be introduced by monotonicity somewhere higher in the proof. Therefore the proof has the form on the top. Because no rule other than monotonicity introduces or eliminates $f_\bullet$ in a positive context, we know that $f_\bullet(a_1,\dots,a_n)$ stays in tact in the proof part represented by the dots, so we can replace it by any formula. Therefore we can rewrite the proof in the following way:

$$\dfrac{\dfrac{\dfrac{a'_k \to a_k \text{ for } k \neq i \qquad a_i \to a'_i}{f^i_\to(a'_1,\dots,a'_n) \to f^i_\to(a_1,\dots,a_n)}\,\text{Mon}}{\vdots}}{\dfrac{b \to f^i_\to(a_1,\dots,a_n) \qquad\qquad f^i_\to(a_1,\dots,a_n) \to c}{b \to c}\,\text{Cut}}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{a'_k \to a_k \text{ for all } k \qquad f^i_\to(a_1,\dots,a_n) \to c}{f^i_\to(a'_1,\dots,a'_n) \to c}\,P_\bullet}{\vdots}$$
$$b \to c$$

We eliminate a cut of complexity $|b| + (\sum_{1\le i\le n} |a_i| + 1) + |c|$, and add n cuts of maximal complexity $\max_{1\le k\le n}(\sum_{1\le i\le k} |a_i| + \sum_{k\le i\le n} |a'_i|) + |c|$. Because no rule (except cut, which does not occur in the dotted part) has more atoms in one of the hypotheses than in the conclusion, it holds that $\sum_{1\le i\le n} |a'_i| \le |b|$. Therefore the new cuts have indeed lower complexity.

Case 2b: The cut formula of the uppermost cut has the form $f^i_{\rightarrow}(a_1, \ldots, a_n)$. As the $f_{\rightarrow}$-connective can only be introduced in a negative context by monotonicity, the antecedent of the right premise must be introduced by monotonicity somewhere higher in the proof. Therefore the proof has the form on the top. Because no rule other than monotonicity introduces $f_{\rightarrow}$ in a negative context, we know that $f^i_{\rightarrow}(a_1, \ldots, a_n)$ stays in tact in the proof part represented by the dots. Therefore we can replace it by any formula, so we can rewrite the proof as follows:

$$\dfrac{b \to f^i_{\rightarrow}(a_1, \ldots, a_n) \qquad \dfrac{\dfrac{a'_k \to a_k \text{ for } k \neq i \qquad a_i \to a'_i}{f^i_{\rightarrow}(a_1, \ldots, a_n) \to f^i_{\rightarrow}(a'_1, \ldots, a'_n)}\text{Mon}}{\vdots}}{b \to c}\text{Cut}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{a'_k \to a_k \text{ for } k \neq i \quad a_i \to a'_i \qquad b \to f^i_{\rightarrow}(a_1, \ldots, a_n)}{b \to f^i_{\rightarrow}(a'_1, \ldots, a'_n)}P_{\rightarrow}}{\vdots}$$

$$b \to c$$

Again we can check that the new cuts have lower complexity.

Case 2c-d: The cut formula of the uppermost cut has the form $g_{\bullet}(a_1, \ldots, a_n)$ or $g_{\rightarrow}(a_1, \ldots, a_n)$. Those cases are symmetric under ∞. Note in particular that $g_{\bullet}$ can never be introduced or eliminated by Grishin rules at all, and $g_{\rightarrow}$ can never be introduced or eliminated by Grishin rules in a positive context.

□

5 Group-theoretic properties

Now let us consider the symmetries of our system from a group-theoretic perspective, and try to extend theorems 5, 6, 8 and 9. Let us first consider the group with the elements $\bowtie(i, j)$ such that $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$, and function composition as operation. To show that this is indeed a group, we check the required properties. We can easily see that $\bowtie(i, j) \circ \bowtie(i', j') = \bowtie(i + i' \mod n, j + j' \mod n)$, so the set is closed under the group operator. Furthermore composition is associative. The element $\bowtie(0, 0)$ functions as identity element, because it is equal to $Z_{I,I}$ where I is the identity permutation. Finally the inverse of $\bowtie(i, j)$ is $\bowtie(-i, -j)$, as $\bowtie(i, j) \circ \bowtie(-i, -j) = \bowtie(-i, -j) \circ \bowtie(i, j) = \bowtie(0, 0)$. It turns out that the group in question is isomorphic to $\mathbb{Z}_n \times \mathbb{Z}_n$, with $\bowtie(1, 0)$ and $\bowtie(0, 1)$ as generators. This is indeed a generalization of the group $\mathbb{Z}_2 \times \mathbb{Z}_2$ that we found for $n = 2$ in theorem 6.

With $\infty\bowtie(i, j)$ we mean the operation consisting of first applying $\bowtie(i, j)$, and then applying ∞. Now we consider the group with the elements $\bowtie(i, j)$ and $\infty\bowtie(i, j)$ such that $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$ and composition as operation. We will abbreviate $\infty\bowtie(0, 0)$ with ∞. We have closure because $\bowtie(i, j) \circ \bowtie(i', j') = \bowtie(i + i' \mod n, j + j' \mod n)$, $\infty\bowtie(i, j) \circ \bowtie(i', j') = \infty\bowtie(i + i' \mod n, j + j' \mod n)$, $\bowtie(i, j) \circ \infty\bowtie(i', j') = \infty\bowtie(i' + j \mod n, j' + i \mod n)$, and $\infty\bowtie(i, j) \circ \infty\bowtie(i', j') = \bowtie(i' + j \mod n, j' + i \mod n)$. Of course composition is associative. As an identity element we have $\bowtie(0, 0)$, because it is equal to $Z_{I,I}$ where I is the identity permutation. The inverse element of $\bowtie(i, j)$ is $\bowtie(-i, -j)$, and the inverse of $\infty\bowtie(i, j)$ is $\infty\bowtie(-j, -i)$.

The group presentation is as follows: $\langle x, y, z; x^n = e, y^n = e, z^2 = e, xy = yx, xz = zy, yz = zx \rangle$, with $\bowtie(1, 0)$, $\bowtie(0, 1)$ and ∞ as generators. This group is a generalization of D_4, which we obtained for the case $n = 2$ in theorem 9. We can see this as a group acting on two planes. The operation ∞ maps the upper plane to the lower plane and vice versa. The operation $\bowtie(1, 0)$ can be seem as a translation in the x direction on the lower plane, and in the y direction on the upper plane, while $\bowtie(0, 1)$ is a translation in the x direction on the lower plane, and in the y direction on the upper plane.

6 Discussion and further work

In this paper, we presented Lambek–Grishin calculus for n-ary connectives $\mathbf{LG}^{\emptyset}_{\mathbf{n}} + \mathbf{IV}$. This calculus can be seen as a generalization over binary and unary Lambek–Grishin calculus ($\mathbf{LG}^{\emptyset}_{\mathbf{2}} + \mathbf{IV}$ and $\mathbf{LG}^{\emptyset}_{\mathbf{1}} + \mathbf{IV}$)

and is therefore at least mildly context-sensitive. In this paper, we proved that there is a cut-free presentation of the calculus, giving us decidability. We also investigated the symmetries of $\mathbf{LG}^{\emptyset}_{\mathbf{n}} + \mathbf{IV}$ by making use of group theory.

The extension of Lambek–Grishin calculus to n-ary arity gives rise to many new questions. First, although we know $\mathbf{LG}^{\emptyset}_{\mathbf{n}} + \mathbf{IV}$ generates non-context-free languages, no upper bound on the generative complexity is known. Secondly, it will be interesting to study the Kripke semantics of the new calculus: it is not yet known whether the results from [7] for $\mathbf{LG}^{\emptyset}_{\mathbf{2}} + \mathbf{IV}$ and [3] for $\mathbf{LG}^{\emptyset}_{\mathbf{1}} + \mathbf{IV}$ can be extended to $\mathbf{LG}^{\emptyset}_{\mathbf{n}} + \mathbf{IV}$. Furthermore, it has been shown that each postulate in the binary calculus can be presented in six interderivable forms [4]. However, in the unary calculus only one form exists for each postulate. It is unknown how the number of interderivable postulates relates to the arity of the connectives. Furthermore, it is useful to try to extend the continuation semantics from Bernardi en Moortgat [1] to the n-ary calculus. Finally, in this paper, we only considered languages with connectives of equal arity. It also will be interesting to see what happens when we combine connectives of different arities.

Acknowledgements

This paper was written as part of my master thesis project. I would like to thank Michael Moortgat and Vincent van Oostrom for supervising this project.

References

[1] R. Bernardi and M. Moortgat. Continuation semantics for symmetric categorial grammar. In D. Leivant and R. de Quieros, editors, *Proceedings WoLLIC'07*, pages 53–71. LNCS 4576. Springer, 2007.

[2] W. Buszkowski. Logical foundations of ajdukiewicz–lambek categorial grammars. *PWN*, 1989.

[3] A. Chernilovskaya. The lambek-grishin calculus for unary connectives. In *Proceedings of the Workshop Symmetric calculi and Ludics for the semantic interpretation, 20th European Summer School on Logic, Language, and Information (ESSLLI)*, Hamburg, 2008.

[4] V. N. Grishin. On a generalization of the ajdukiewicz-lambek system. In A.I. Mikhailov, editor, *Studies in Non-classical Logics and Formal Systems*, pages 315–343. Nauka, Moscow, 1983.

[5] R. Huybregts. The weak inadequacy of context-free phrase structure grammars. In M. Trommelen G. J. de Haan and W. Zonneveld, editors, *Van Periferie Naar Kern*, pages 81–99. Foris Publications, Dordrecht, 1984.

[6] G. Jäger. Residuation, structural rules and context freeness. *J. of Logic, Lang. and Inf.*, 13(1):47–59, 2004.

[7] N. Kurtonina and M. Moortgat. Relational semantics for the lambek-grishin calculus. In G. Penn M. Kracht and E. Stabler, editors, *Proceedings of the 10th Mathematics of Language Conference, UCLA Working Papers in Linguistics*, Los Angeles, 2007.

[8] J. Lambek. On the calculus of syntactic types. In R. Jacobsen, editor, *Structure of Language and its Mathematical Aspects*, Proceedings of Symposia in Applied Mathematics, XII, pages 166–178. American Mathematical Society, 1961.

[9] M. Moortgat. Symmetries in natural language syntax and semantics: The lambek-grishin calculus. In D. Leivant and R. de Quieros, editors, *Proceedings WoLLIC '07*, pages 264–284. LNCS 4576. Springer, 2007.

[10] M. Moortgat and R.T. Oehrle. Proof nets for the grammatical base logic. In V. M. Abrusci and C. Casadio, editors, *Dynamic Perspectives in Logic and Linguistics: Roma Workshop IV*, pages 131–144, Roma, 1999.

[11] R. Moot. Lambek grammars, tree adjoining grammars and hyperedge replacement grammars. In *Proceedings of the TAG+ Conference*. HAL - CCSD, 2008.

Collective Intelligent Wireless Sensor Networks

Mihail Mihaylov [a] Ann Nowé [a] Karl Tuyls [b]

[a] *Vrije Universiteit Brussel, Pleinlaan 2, Brussel, BE,*
{*mihail.mihaylov; ann.nowe*} [AT] *vub.ac.be*
[b] *Technische Universiteit Eindhoven, Postbus 513, Eindhoven, NL, k.p.tuyls* [AT] *tue.nl*

Abstract

In this paper we apply the COllective INtelligence (COIN) framework of Wolpert et al. to Wireless Sensor Networks (WSNs) with the aim to increase the autonomous lifetime of the network in a decentralized manner. COIN describes how selfish agents can learn to optimize their own performance, so that the performance of the global system is increased. WSNs are collections of densely deployed sensor nodes that gather environmental data, where the main challenges are the limited power supply of nodes and the need for decentralized control. To overcome these challenges, we make each sensor node adopt an algorithm to optimize its own energy efficiency, so that the energy efficiency of the whole system is increased. We introduce a new private utility function that will measure the performance of each agent and we show that nodes in WSNs are able to develop an energy saving behaviour on their own, when using the COIN framework.

1 Introduction

An increasingly popular approach for environmental and habitat monitoring is the use of Wireless Sensor Networks (WSNs) [2]. The nodes in a WSN are limited in power, processing and communication capabilities, which requires that they optimize their activities, in order to extend the autonomous lifetime of the network. A complicating factor is communication, because some nodes fall outside the transmission range of the base station, or can belong to different stakeholders, serving various purposes, thus rendering the common centralized approach inapplicable for large networks.

In this paper we use the MacCOIN approach, proposed by Beyens et al. [1], to optimize the energy efficiency of a WSN by making nodes (hereby regarded as agents) learn energy-saving schemes by themselves in a distributed manner. This is achieved by applying the COllective INtelligence (COIN) framework of Wolpert et al. [7, 8, 6]. The idea behind this approach is that agents use an algorithm to optimize their performance, based on a certain performance measure, defined by a utility function. Whereas the contribution of [1] remains conceptual, we apply the approach to simulated sensor networks in this paper. Moreover, we propose two different algorithms and a utility function that will help agents achieve energy efficiency and thus extend the autonomous lifetime of the WSN.

The outline of the paper is as follows: Section 2 presents the background of our approach by describing the basics of a wireless sensor network and the MAC communication protocol. Section 3 describes the idea behind collective intelligence and its application to the energy efficiency optimization of nodes. Section 4 shows the results of our experiments and some discussion about its applicability in real life settings. Lastly, Section 5 presents our conclusions from this research and suggests some areas for improvement in the future.

2 Background

In this section we describe the basics of a Wireless Sensor Network and the MAC communication protocol. Subsection 2.1 shows the challenges of WSN communication and the design criteria for an energy efficient network. Subsection 2.2 describes the MAC protocol used in our research and the operation of a single node in the WSN.

2.1 Wireless Sensor Networks

A Wireless Sensor Network is a collection of densely deployed autonomous devices, called sensor nodes, that gather environmental data with the help of sensors. The untethered nodes use radio communication to transmit sensor measurements to a terminal node, called the sink. The sink is the access point of the observer, who is able to process the distributed measurements and obtain useful information about the monitored environment. Sensor nodes communicate over a wireless medium, by using a multi-hop communication protocol that allows data packets to be forwarded by neighbouring nodes to the sink. This concept is illustrated in Figure 1. The environmental or habitat monitoring is usually done over a long period of time, taking into account the latency requirements of the observer.

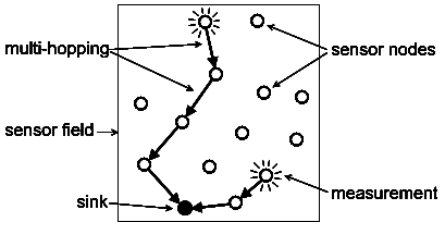


Figure 1: Wireless Sensor Network

The WSN can vary in size and topology, according to the purpose it serves. The sensor network is assumed to be homogeneous where nodes share a common communication medium (e.g. air, water, etc.). We further assume that the communication range is equal in size and strength for all nodes. They have a single omnidirectional antenna that can only *broadcast* a message, delivering it to all nodes in range. In our network, sensor nodes can neither vary their transmission power, nor are they able to estimate their distance from the transmitting node by measuring the signal strength – such features are not generally available in sensor nodes and therefore are not considered here. The motivation to use such simple devices is to reduce the overall cost of nodes and to keep our solution applicable to the most general sensor network.

In this paper we use the COIN framework to make the selfish and computationally bounded agents optimize their own performance in order to achieve the global objective of energy efficiency. The framework assumes that communication between the agents is limited and that central control is not possible [8]. We further require that the communication protocol considers not only energy efficiency, but also scalability and fault tolerance, so that our approach is able to adapt to a dynamic topology, where nodes may fail or new nodes may be added to the system. The communication protocol, therefore, constitutes an important part of the WSN design.

2.2 The MAC Protocol

The Medium Access Control (MAC) protocol is a data communication protocol, concerned with sharing the wireless transmission medium among the network nodes. Typical MAC protocols, used by ad-hoc networks, cannot be applied to WSNs, due to a number of differences between the two types of networks. Some differences include the large number and density of sensor nodes in a WSN, compared to the nodes in ad-hoc networks; the frequently changing topology of sensor nodes and their power constraints, etc.

In our research we use a Time Division Multiple Access (TDMA) protocol that divides the signal into frames, where each frame is further divided into time slots. Each node independently determines its schedule, i.e. the amount of time slots it will spend in one of its four modes of operation: transmit, listen, receive and sleep. Those modes are defined as follows:

- A node goes in **transmit** mode, when it starts to send a message through the channel. The omnidirectional antenna of the node broadcasts the message to *all* nodes in range, which we call neighbours (or neighbouring nodes), but the message itself is addressed to only *one* of them. Neighbouring nodes that receive a broadcasted message, which is not addressed to them, will simply discard it.
- When in **listen** mode, the sensor node is actively listening for broadcasts in the medium. When a signal is detected, the node goes in receiving mode.
- A sensor node is in **receive** mode when receiving a message from a neighbouring node. When the whole message is received, the node goes in one of the three other modes.
- When a node is in **sleep** mode, its radio transceiver is switched off and therefore no communication is possible. Nevertheless, the node continues its sensing and processing tasks.

The TDMA protocol allows nodes to synchronize their schedules prior to communication and thus avoid collisions and overhearing – typical sources of energy waste. The amount of wasted energy is very much dependent on the hardware of the node. Therefore we will provide only relative values for the energy consumption of the above four modes of operation (for exact values, see [5]):

- transmitting is two times more expensive (in energy consumption) than listening or receiving;
- listening and receiving consume nearly the same energy;
- sleeping is ten times cheaper than listening or receiving.

Given these energy consumption rules, it is clear that in order to save more energy, a node should sleep more. However, when sleeping, the node is not able to send or receive any messages, therefore it increases the latency of the network, i.e. the time it takes for messages to reach the sink. On the other hand, a node does not need to listen to the channel when no messages are being sent, since it loses energy in vain. As a result, nodes should learn on their own the number of time slots they should spend listening and sleeping within a frame. For example, nodes far away from the sink may learn to sleep more, since they will have fewer messages to forward, while nodes close to the sink should learn to listen more, because the workload near the sink is usually heavier. Learning to optimize nodes' own schedules will ensure good energy efficiency of the network, while preserving the latency requirements of the observer. The MAC protocol should therefore support the exchange of additional information, necessary for the algorithms for optimization. It is clear that the amount of this information within message packets should be kept as little as possible, to minimize the energy waste by control packet overhead. Since it is not our aim here to provide detailed description of the protocol used, we refer the interested reader to [3].

3 Methodology

In this section we present the collective intelligence framework and then apply it to the energy conservation problem of WSNs. In Subsection 3.2 we define a performance measure function for the energy efficiency of nodes and let them use an algorithm to optimize their efficiency, based on this function.

3.1 COIN Framework

The emerging science of COllective INtelligence (COIN) considers how to design large multi-agent systems, where selfish agents learn to optimize a private utility function, so that the performance of a global utility is increased. The main difficulty in COIN is that there is little to no centralized control over the behaviour of the computationally bounded agents. Each agent is guided by its own *private utility*, while the performance of the collective is measured by the *global utility* (or *world utility*). Thus, the main challenge in the COIN framework is to define a suitable private utility function for the individual agents that will lead to an effective emergent behaviour. Several utility functions for different problems have been proposed in the past [7], but they cannot be applied to the energy conservation problem of wireless sensor networks, due to the limited information available to nodes. Agents in a WSN can obtain only local information from surrounding nodes, due to their small transmission range. In the next Subsection we propose a new utility function that will lead to a global energy efficient behaviour.

3.2 MacCOIN

To find a good private utility function that will measure the performance of agents, we first introduce what Energy Efficiency (EE) of a single agent is. Then we formulate the utility function of an agent and explain how it can be obtained from local information. Lastly, we present two algorithms for private utility optimization, which will make nodes find an energy efficient solution by themselves.

3.2.1 Energy Efficiency

An agent is energy efficient when it minimizes most of the major sources of energy waste in WSN communication – idle listening, overhearing, unsuccessful transmissions and control packet overhead, while maximizing sleep mode and taking into account the latency requirement of the observer. In this paper we will not consider the latency requirement of the user *directly*, because the latency of a packet is only known

when it arrives at the sink. Instead, we include a latency component in the function for EE of an agent. This component "penalizes" the agent for each frame that it holds a packet in its queue. Additionally, we specify the maximum number of frames during which a data packet is allowed to stay in the queue of the sensor node. More specifically, if a packet remains in the queue of a node for more than 3 frames, that packet is dropped and thus will not be delivered to the sink, because it has expired. This value depends on the application of the WSN. It has been experimentally chosen to stimulate faster forwarding by penalizing the node for dropped packets, while still maximizing the number of delivered messages in a network, designed for environmental monitoring. For a formal description of the energy efficiency of agents, we refer the interested reader to [1]. The motivation to consider latency in such an indirect manner is that we focus rather on investigating whether nodes *can* develop energy saving schemes on their own and to examine the trade-off between efficiency and latency for different algorithms for optimization. After introducing the energy efficiency of a single agent, we will motivate the need for a suitable private utility function and then will proceed with its formulation.

3.2.2 Private Utility Function

It is easy to show that if agents try to increase simply their *own* energy efficiency, that will not lead to high global efficiency, because agents will tend to sleep more (since that is the cheapest mode) and hence – higher latency of the network[1] and more unsuccessful transmissions of neighbouring nodes. Therefore, individual agents should also consider other agents in the system when optimizing their own behaviour. We believe that if each agent "cares about the rest" that will improve the performance of the whole system. To achieve that, we introduce the concept of an Effect Set (ES) of a node, which is the set of all nodes that are influenced by the actions of that single node. In other words, the ES of agent i is the set N_i of nodes that lie on the possible message paths between agent i and the sink, since those agents have to forward the messages of agent i.[2] Thus, the energy efficiency of all agents in N_i is directly dependent on the actions of agent i. Therefore, if each agent seeks to increase not only its own efficiency, but also the efficiency of its ES, our hypothesis is that this will lead to higher energy efficiency of the whole system. For this reason, we set the private utility of each agent to its Effect Set Energy Efficiency (ESEE). We define the ESEE of agent i in frame f as

$$\text{ESEE}_{i,f} = \frac{\text{EE}_{i,f} + \sum_j \text{EE}_{j,f}}{|N_i| + 1} \qquad \forall j \in N_i$$

where $\text{EE}_{i,f}$ is the energy efficiency of agent i in frame f and $|N_i|$ is the number of agents in the effect set of agent i. In other words, the ESEE of an agent is the average EE of all the agents in its effect set, together with its own efficiency. Thus, agents will try to increase the value of their ESEE by optimizing their own behaviour.

The idea behind the effect set of a node is based on the work of Wolpert et al. [7], who introduced this concept for their Wonderful Life private utility function. However, they use this concept in a different way, namely to "clamp" the actions of agents in an ES in order to determine whether the system performs better without them. The Wonderful Life utility is further explained in Subsection 4.3.

One challenge for our private utility function is that nodes cannot compute their ESEE *directly*, because to do so, they would have to obtain the efficiency of each agent in N_i and compute the average. Obtaining the energy efficiency of distant nodes is merely not possible due to the small transmission range of nodes. Instead, they should only *estimate* the efficiency of their ES. To achieve that, nodes have to send information about the efficiency of all nodes in range, i.e. their neighbourhood, to their neighbours. This information is easy to obtain for each node, because neighbouring nodes, by definition, lie within transmission range. Starting from the neighbours of the sink, each node takes the average of the efficiencies of its own neighbourhood and the ES of its neighbours[3] (which in turn do the same, while excluding already counted nodes), to obtain its own ESEE. Thus, the ESEE of agent i is the average energy efficiency of all the neighbourhoods between agent i and the sink, i.e. all the nodes in N_i. We will now show how each agent can learn to optimize its ESEE.

[1] unless the latency component "penalizes" nodes too much, in which case they will try to forward packets very often to decrease latency, leading to high control packet overhead and higher chance of collisions and overhearing.

[2] The set N_i could be different in size for each node i, depending on the topology of the network and the position of agent i in it.

[3] only those neighbours that lie on the possible message paths towards the sink, i.e. only those nodes in i's neighbourhood that are closer to the sink.

3.2.3 Algorithms for ESEE Optimization

Besides on its hardware, the energy consumption of a node is also dependent on its position in the WSN. Nodes, closer to the sink have to forward more messages and therefore need to listen more, while those, far away from the sink, could spend more time sleeping. For this reason, the behaviour of agents cannot be the same for all (e.g. all listen and sleep the same amount of time in a frame). Each node needs to *learn* what behaviour is energy efficient both for itself and for its ES. To achieve that, we make nodes adopt an algorithm for optimization in order to increase the payoff of their private utility function, i.e. to optimize their ESEE. We tested the effect of two different algorithms on the performance of nodes. Each node compares the ESEE of the last frame to the one before that in order to determine whether its efficiency has dropped or increased. This learning is done on-line, since the algorithms adapt to the topology of the network and the traffic pattern, which typically cannot be known in advance in order to train them off-line. Based on their private utility function and the algorithm they implement, the nodes will take different actions.

The first algorithm lets nodes listen more, if their efficiency drops. The motivation behind it is to decrease the latency of the network and the energy waste of unsuccessful transmissions, caused by nodes, trying to send messages to sleeping neighbours. This algorithm is a simple decision rule, saying that if the ESEE of a node drops, regardless of the amount, that node should increase the time for listening by a fixed amount of time slots within a frame. On the other hand, if the ESEE of the node increases, the decision rule specifies that the node should sleep more, because nodes in its ES are "already doing well" and therefore it can safe more energy for itself by sleeping.

The second algorithm is a learning function, which aims to decrease the energy waste of idle listening and overhearing by letting nodes sleep more, if their ESEE drops. Nodes try to optimize this learning function by changing their listening and sleeping intervals according to the change in the payoff of their private utility. The faster this payoff decreases, the more time slots the node will sleep in order to minimize idle listening and overhearing. Increasing the ESEE will result in more listening, so that the latency is reduced.

Both algorithms have opposite effect on the nodes' behaviour, because one cannot know in advance what the *major* source of energy waste in a WSN is. Therefore, we seek to investigate this issue by designing two different algorithms, aimed at minimizing different sources of energy waste, and observing how the ESEE of nodes change. Additionally, those algorithms for optimization will have different effect on nodes in different parts of the WSN, depending on node density and message rate. In this way, nodes, close to the sink might learn to listen more, while distant nodes might find that more sleeping optimizes their performance, based on the payoff of their private utility function.

4 Results

4.1 Experimental Setup

The wireless networks, used in the experiments, were composed of nodes, aligned in a rectangular grid structure, rather than being randomly distributed in the environment. Although this is rarely applied in practice, we use such a simple structure to test the *theoretical application* of the COIN framework, rather than to propose an "out-of-the-box" practical solution. We believe that a grid structure is suitable for the purposes of this research. The transmission range of all nodes is 1.5 grid units in radius, therefore, a node can have at least 3 neighbours, when at the "corner" of the grid, and at most 8, when in the inner part of the grid. In other words, all immediate neighbouring nodes in the grid are also neighbours in the network. Initially, all nodes start synchronously to listen and sleep for equal amount of time slots in a frame. One frame consists of 10 time slots, which means that the periodic listening occurs in the first 5 slots and the sleeping – in the last 5 slots of each frame for all nodes. Agents forward messages only to lower hop neighbours, i.e. to nodes that are closer to the sink, than they are. If more than one such nodes exist, the sender will choose the one that has processed fewer messages in the past frames. The information about the amount of processed messages is shared between neighbouring nodes, so that each node keeps track of the number of processed messages of its neighbours. We assume that this simple routing algorithm performs better than sending a message to a random neighbour. Lastly, we perform our experiments on a network consisting of 100 nodes, for the duration of 5000 frames, in which each node makes 4 measurements in the environment. We assume that measurements may have variable length (e.g. due to compression or information fusion with measurements of neighbouring nodes [4]) and therefore one measurement may take up more than one message packets.

4.2 Experiments

We evaluated three experimental settings, designed to test the performance of nodes:

0 Nodes use **no algorithm** to optimize their private utility function. They are simply following their fixed periodic schedules with equal number of time slots for listening and sleeping.[4] This scenario serves only as the base line for the two algorithms for optimization.

1. Nodes use a **decision rule** to determine the amount of time slots to sleep and to listen in their periodic schedules. A decreasing payoff of their private utility function (irrelevant of the magnitude) results in more listening, while an increasing payoff makes the node sleep more. The motivation behind this algorithm is to decrease the latency of the network and the energy waste of unsuccessful transmissions.

2. Nodes use a **learning function** to find a periodic schedule that will ensure high values of energy efficiency. The more the ESEE of a node is decreasing, the more the node will tend to sleep, while increasing ESEE results in more listening. The purpose of this algorithm aims to decrease the energy waste of idle listening and overhearing.

In each experiment we measured the minimum, average and maximum battery life (i.e. mean and standard deviation) among all nodes and the latency of the network. The average battery life indicates how fast nodes dissipate their energy across time. The steeper this value drops over time, the less energy efficient the network is. The minimum and maximum battery life show the difference of energy levels between the least and the most efficient nodes respectively. The closer these values are around the average battery life, the better the efficiency of the algorithm is, because that would indicate a uniform dissipation of energy across the network. The latency of the network is expressed in terms of received messages, which signify the number of messages that the sink has received throughout the simulation runtime (5000 frames) as a part of all generated messages within that time. Here the intuition of latency is reversed, because higher number of received messages means low latency. When this value is 100%, it means that all generated messages are received by the sink within those 5000 frames.

The results of each experiment are averaged over 1000 runs with the same settings, but with a different random seed and position of the sink in the network. Figure 2 shows the performance of the network when nodes do not optimize their behaviour. All nodes simply listen and sleep for equal amount of time slots in each frame. However, when nodes use the above mentioned decision rule to improve the efficiency of their ES, the sensor network increases its performance significantly, as shown in Figure 3. In the latter case, the average remaining battery in the end of the simulation is 83%, while with no optimization, nodes waste more energy and therefore have only 60% remaining battery after the 5000 time frames. Recall that this decision rule aims to decrease the energy waste of unsuccessful transmissions by letting nodes *listen* more if their efficiency drops. However, we observed that nodes nevertheless tend to sleep more, since sleeping is the cheapest mode. This behaviour improves their efficiency and therefore, according to the decision rule, leads to even more sleeping (cf. subsection 3.2.3). These results imply that unsuccessful transmissions do not constitute a big source of energy waste, compared to the energy savings during sleep mode, and therefore nodes improve their efficiency more by sleeping than by trying to minimize unsuccessful transmissions. The drawback of this algorithm is that it causes nodes to sleep much and therefore increases the latency of the whole network. This can be seen in Figure 4, which shows the percent of received messages by the sink within 5000 frames for the three experiments. Although the decision rule is able to make nodes save more energy, it decreases the response time of the network significantly, because the results showed that almost all nodes increase their sleeping duration, rather than appropriately adapting their periodic schedules to the message flow in the network.

The second algorithm for optimization, i.e. the learning function, shows a comparable improvement in energy efficiency to the first (see Figure 5). This learning function aims to decrease the energy waste of idle listening and overhearing by letting nodes *sleep* more, if their ESEE drops. The difference to the above mentioned decision rule is that if here the efficiency of the nodes increases, they will start to listen more in order to decrease the latency of the network. The result of the latter behaviour can be seen in Figure 4. Despite the comparable high energy savings to the first algorithm, the learning function performs significantly better in terms of latency. This is because nodes near the sink usually have to process more messages and thus, according to the algorithm, learn to increase their payoff by listening more, while nodes

[4] Although we say there is "no actual optimization", nodes still save nearly 50% more energy, compared to conventional networks, because in our network they sleep already half of their lifetime.

far from the sink learn to avoid idle listening by spending more time sleeping, than they do with the decision rule. Therefore, on average, the energy savings with the second algorithm are comparable to the first, but the latency of the system using the learning function is far less, because nodes learn to adapt better to the message flow in the network, i.e. to minimize not only efficiency, but latency as well. In Figure 4 one can see that 78% of all generated messages are received by the sink within 5000 frames, when nodes optimize their behaviour using the second algorithm. Thus, the latency with the learning function is three times less than with the decision rule, while both algorithms show similar performance in terms of energy efficiency. This behaviour also explains the lower minimum battery level of the network (see Figure 5). This lower minimum is caused by all nodes near the sink, because they dissipate more energy by trying to minimize latency, i.e. by spending more time slots on listening and thus forwarding all the messages of other nodes.

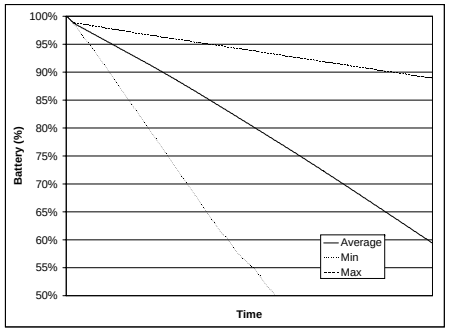


Figure 2: Battery life throughout simulation; 100 nodes, no algorithm for optimization

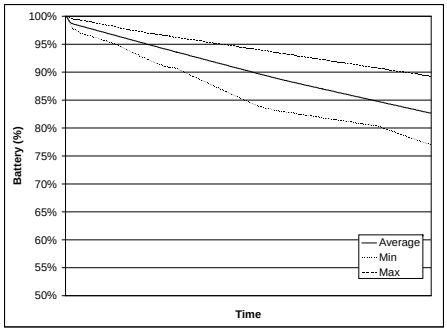


Figure 3: Battery life throughout simulation; 100 nodes, optimization using a decision rule

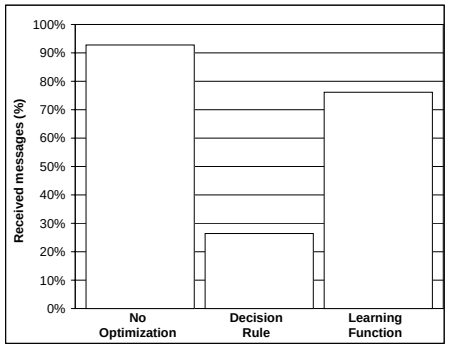


Figure 4: Amount of received messages by the sink within the duration of the simulation (5000 frames)

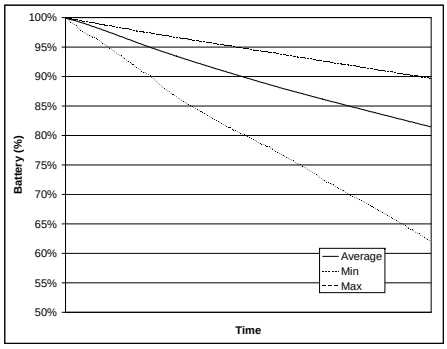


Figure 5: Battery life throughout simulation; 100 nodes, optimization using a learning function

4.3 Discussion

These experiments demonstrated the challenge of the COIN framework in finding an appropriate private utility function and an algorithm for optimization to make nodes achieve the designer's goal on their own in a decentralized fashion. We were able to show that when the algorithm is chosen carefully, it leads to a desirable outcome, as it was the case with the learning function. It was able to achieve high energy efficiency, while not sacrificing latency a lot.

In the above experiments we used the ESEE as a private utility function for each node. One advantage of this function is that it makes each agent consider the effect of its own behaviour on the other agents in the system. In this way, when agents learn to optimize their ESEE, they in fact learn to reduce the negative effect of their actions on the system. In other words, by using this private utility function, we align the agents' goal of improving one's own energy efficiency with the system goal of increasing the autonomous lifetime of the whole network. Additionally, the above two algorithms adapt well to adding or removing nodes from the system when the necessary information for the update propagates through the network. The

disadvantage is that updating the value of ESEE happens rather slow for large networks, due to the large number of messages that need to be exchanged.

Alternatively, we have tested a similar private utility function, proposed in the work of Wolpert et al. [7], namely the Wonderful Life Utility (WLU). This function lets each node estimate "how the world would look, if that node had never existed". In other words, agents try to determine their effect on the whole system by computing the performance of the network if they are removed from it. One disadvantage of the WLU is that each node needs to know the energy efficiency of the *whole* network and of the nodes that are *not* part of its ES, in order to determine whether the system will perform better without it. These requirements increase not only the delay of the payoff, but also the amount of information that needs to be transmitted across the network, so that nodes can compute the WLU. We, therefore, found this private utility function less suitable for this research.

5 Conclusion

In this paper we applied the COllective INtelligence (COIN) framework to the energy conservation problem of Wireless Sensor Networks (WSN) in order to prolong the autonomous lifetime of the network, while taking into account the latency requirement of the user. We let individual nodes adopt an algorithm to optimize their own behaviour, so that the energy efficiency of the whole system is increased. Each node seeks to improve not only its own energy efficiency, but also the efficiency of the nodes it affects, which will ensure that the agents' goal is aligned with the system goal of higher energy efficiency. The two algorithms, presented in this paper, are compared to a setting where nodes do not optimize their behaviour at all. The results show that when sensor nodes learn to optimize their actions, their battery consumption is reduced by more than 70%, compared to conventional ad-hoc networks. One drawback to this approach is that it comes at the cost of increased latency. However, in most WSN applications for environmental or habitat monitoring, latency is not that crucial and therefore it can be sacrificed to some extent in order to prolong the lifetime of the network to allow for longer observations.

We are currently focusing on extending the approach, presented in this paper, to make it suitable for a larger set of WSN applications, where the network will adapt to the latency requirement of the user directly. Future work involves improving the routing protocol, experimenting with different network topologies and testing additional algorithms for optimization to obtain a better trade-off between energy efficiency and latency.

References

[1] Pieter Beyens, Karl Tuyls, Kris Steenhaut, and Ann Nowe. Maccoin: Medium acces control for wireless sensor networks based on collective intelligence. In *Proceedings of GTDT'04*, pages 1–12, 2004.

[2] Jean Carle and David Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *IEEE Computer Society*, 47(2):40–46, 2004.

[3] Mihail Mihaylov. Computational Mechanism Design for Wireless Sensor Networks. Master's thesis, Maastricht University, The Netherlands, March 2008.

[4] Alex Rogers, Rajdeep K. Dash, N. R. Jennings, Steve Reece, and Stephen Roberts. Computational mechanism design for information fusion within sensor networks. In *Ninth International Conference on Information Fusion*, 2006.

[5] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings Of The First International Conference On Embedded Networked Sensor Systems*, pages 171–180, Los Angeles, California, USA,, 2003.

[6] David Wolpert and Kagan Tumer. *Collectives and the Design of Complex Systems*, chapter A Survey of Collective Intelligence, pages 1–42. Springer-Verlag, 2004.

[7] David H. Wolpert. Collective intelligence. Technical report, NASA Ames Research Center, 2003.

[8] David H. Wolpert and Kagan Tumer. Collective intelligence, data routing and braess paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002.

Effects of Goal-Oriented Search Suggestions

James Mostert Vera Hollink

University of Amsterdam, Kruislaan 419, 1098 VA Amsterdam

Abstract

Many search engines help users to refine their search queries by showing a number of search suggestions, in addition to the search results. These suggestions can be used as queries in the next step of the search process. In this work we present a method to classify automatically the goal of a user as navigational (finding one specific web site), 'hot' (finding information that has recently become popular) or other. We use this classification to offer search suggestions that match the user's current search goal. In a large-scale experiment on a popular, real life search engine, we compare the usage of goal-specific suggestions to a one-size-fits-all approach. We show that taking the users' goals into account can significantly increase the use of search suggestions. Moreover, it improves the effectiveness of the search sessions that follow the use of the suggestions.

1 Introduction

A remaining problem for internet search engines is the fact that many users do not search optimally from the perspective of information retrieval: search queries generally contain few words, Boolean operators are seldom used, users enter few reformulations in their search session, and they do not often go further than the first results page [14].

One of the approaches to reduce this problem is to make use of interactive query expansion: to offer related search suggestions on the results pages, allowing the user to change or refine his search query easily, as well as providing inspiration for a manual reformulation [7]. An example of a search engine with interactive query expansion can be found in figure 1.

A relatively recent addition to research on the usage of search engines revolves around the question why users submit a query. If his goal is known, a search engine can answer a user's need for information better by re-ranking its results or by offering direct links to related web services (e.g., a map service for location queries) [2]. However, the user's motivation is generally not stated literally in his search query.

This research combines interactive query expansion with the identification of user goals. If the implicit user goal can be identified for a certain query, we expect it to be possible to provide search suggestions that are geared towards that user goal. This article focuses on several questions: how can goals of search engine users be identified? Can goal-oriented search suggestions be generated for these goals? And finally, what are the effects of these goal-specific suggestions on the search behaviour of the user? We present methods to identify the user's goal and to generate search suggestions that are more specific to that goal. To answer the last of the questions we performed a large-scale experiment on the Dutch search engine ilse.nl [3]. In the experiment, either generic or goal-specific search suggestions were displayed on the search engine results pages. The logged interaction between the user and the suggestions provided the data with which we could answer our research question.

This paper is organized as follows. Section 2 describes related work. In section 3 the methods for identifying user goals and for generating goal-oriented search suggestions are detailed. Section 4 describes the evaluation setup and results.

Figure 1. Screenshot of search engine ilse.nl with search suggestions below the search box.

2 Related work

The behaviour of search engine users has been researched in several large scale studies. Examples of the findings include: 85% of the search engine users do not look beyond the first results page and 77% of the search sessions only contain one query submission [14, 4]. The research of Rieh and Xie [12] shows that if users *do* reformulate their query, most of those reformulations are parallel movements (another aspect of the same concept) or specifications of the original query (e.g., by adding a term to their query). Based on these results, Rieh and Xie recommend the application of an interactive query expansion system that supports these types of reformulations on search engines.

Finding candidate terms for interactive query expansion can be done in a variety of ways. Beeferman and Berger [1] developed an algorithm that clusters URLs and queries solely on the basis of historical data in a search engine's query log. The biggest advantages of this method are its low CPU cost and its "content ignorance": without knowing anything about the content or meaning of the web pages and search queries, latent relationships can be found between them. Query expansion can also be performed by an analysis of the content of web pages. Recent examples include [5], who used natural language processing to find candidate suggestion in the documents' content, and [8], who used the 'anchor text' of hyperlinks for the identification of possible expansion terms. Content-based methods are computationally expensive, so that users may have to wait a long time before receiving suggestions. A final method for generating relevant search suggestions is through the use of ontologies, although they perform best for relatively small, static and well-defined domains [10].

It should be noted that none of the methods for generating search suggestions take the goal of the search engine user into account. The identification of the user goal helps to gain better insights into user search behaviour [2, 13]. Broder's "taxonomy of web search" [2] has been one of the most influential publications with regard to classifying user goals. His taxonomy consists of three classes: 1) navigational: the user's goal is to reach a website they already have in mind, 2) informational: the user's goal is to find static information on one or more web pages, and 3) transactional: the user aims to reach a site to complete a web-mediated activity (e.g., shopping). Rose and Levinson [13] extended his taxonomy with several subcategories.

If the identification of a search engine user's goal can be performed automatically, search results can be selected and ranked in such a way that the user's information question can be answered more quickly and effectively [11]. Kang and Kim [6] classified search queries into Broder's taxonomy for 'navigational' and 'informational' queries, based on three indicators: document content, link information and URL information. Lee et al. [9] performed the same automatic classification using query log data and link anchors. To our knowledge no attempts have been made to use automatic user goal classification to offer goal-specific interactive query expansion, which is the focus of this paper.

3 Method

3.1 Generic search suggestions

To be able to measure the effects of goal-oriented search suggestions that are placed on the results pages of a search engine, a baseline is needed: data about the effects of search suggestions that are *not* geared towards a specific user goal. To obtain these data, we developed a generic suggestion algorithm. The mechanisms behind this algorithm were inspired by the work of [1], who developed an effective method for generating search suggestions for many queries, while saving on CPU overhead because only query logs are analyzed, instead of fetching and parsing web pages.

The generic algorithm generates its suggestions as follows. From the historical click data in the search engine's query logs, a list of the most clicked URLs is obtained for a given query. Subsequently, for each of those URLs, a list of queries is obtained via which that URL was reached. This process results in a variety of search queries that are located 'in the neighborhood' of the original query. Because the amount of clicks between a query and a URL is known, the neighboring queries can be ranked by popularity. The highest ranked queries are then offered to the user on the search engine results pages as search suggestions. This process is schematically displayed in figure 2.

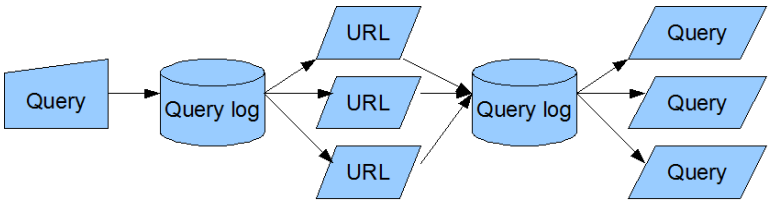


Figure 2. Schematic display of the generic search suggestions algorithm.

The generic algorithm makes use of a query log that contains 28 weeks of click data, an amount that is sufficient to generate suggestions for most queries. To obtain better results, a similarity filter is implemented to remove near-duplicate search suggestions. A pilot study on the query logs of search engine ilse.nl yielded the observation that the generic algorithm produces search suggestions with a high precision (i.e. relevant suggestions), and is able to produce suggestions for many queries. An example of the search suggestions for a query can be found in table 1.

Query	URLs	Candidate suggestions	Final suggestions
belasting *(tax)*	http://www.belastingdienst.nl http://belasting.startpagina.nl/ http://www.de-belasting.nl/ http://www.belastingdienst.nl/particul... http://www.professorfiscus.nl/ http://belasting.finanstart.nl/ http://www.belasting.nl/ …	belastingdienst www.belastingdienst.nl www.belasting.nl belasting.nl belastingdienst.nl wegenbelasting zorgtoeslag huurtoeslag …	belastingdienst *(tax administration)* wegenbelasting *(road tax)* zorgtoeslag *(health care allowance)* huurtoeslag *(rent allowance)* …

Table 1. Intermediary and final output of the generic algorithm for query “belasting” (tax).

3.2 Goal-specific search suggestions

To get some insights into the data set we were working with, we randomly selected 50 users’ search sessions from the query logs of the search engine ilse.nl, and tried to classify them by several existing classification schemes, including those of [2], [11] and [13]. Based on this exploration we chose to develop algorithms for the identification of two user goals:

1. Navigational: the user’s goal is to reach a certain site that the user already has in mind when submitting his[1] query, either because he knows the website exists, or he assumes that it does [2]. This type of query seems to occur with a relatively high frequency among the users of ilse.nl. Previous research on automatic identification of this user goal is available as well.
2. ‘Hot’: for queries that have suddenly become popular, the user goal can be described as finding information about a recently popular subject. According to [15] search engine users pursue this goal frequently. We also expect that these queries can be identified with a relatively high precision due to their unexpected increase in usage.

3.2.1 Navigational queries

The goal of navigational queries is to reach a certain website or page. For example, if a user submits the query ‘startpagina’, it is likely that he wants to visit the website ‘http://www.startpagina.nl’.

Both [6] and [9] developed methods to identify this goal automatically. We chose to base our identification algorithm on the methods of the latter, as they reached similar precision rates using simpler methods. We used the metric ‘click distribution’ to identify navigational queries. For a given query a URL-click-distribution histogram is created from the query log. If 50% or more of the clicks for that query go to one URL, the query is identified as navigational. A typical distribution histogram is shown in figure 3. A pre-condition for this identification process is the presence of sufficient click data to base the

[1] For brevity we use ‘he’ and ‘his’ whenever ‘he or she’ and ‘his or hers’ are meant.

decision on. As a rule of thumb we require at least 50 clicks of historical data to be able to identify a query as navigational or not.

For generating navigational search suggestions, we expected search engine users would be best assisted by suggestions that are specific to the website they are trying to reach. To accomplish that we use the following method: after a query is identified as navigational, the relevant domain for that query is extracted (e.g., 'marktplaats.nl'). We then submit a request to ilse.nl using the "site:" operator, which returns the ten most popular URLs within that domain. For each of those URLs, we look up the queries that have been used in the past to reach those pages, just like the generic algorithm. If the candidate search suggestion does not already contain the original query and the suggestion is not a URL, we precede the search suggestion by the original query, in order to keep the suggestions as domain-specific as possible. For example: if a user submits the query "marktplaats", and one of the resulting search suggestions is "banen", the original query is added, so that the actual search suggestion will be "marktplaats banen". We evaluated the navigational algorithm in a pilot study by comparing its output with the generic algorithm and observed that the resulting search suggestions were in fact more domain-specific. An example of the output of the algorithm can be found in table 2.

A valid question is whether it is actually useful to generate search suggestions for navigational queries: after all, the goal of the user is usually to click on one result. Nevertheless, we expect that search suggestions that are specific to the site that the user wants to reach can help the user to reach the desired page within that site immediately.

Query	Output of generic algorithm	Output of navigational algorithm
markplaats	autoscout	marktplaats banen
	werktuigen	www.marktplaats.nl/motoren
	auto	marktplaats hobby
	stafford	marktplaats computer
	marktplaats startpagina	marktplaats computer hardware

Table 2. Search suggestions for query "marktplaats" by the generic and the navigational algorithm.

3.2.2 'Hot' queries

Search queries that suddenly gain in popularity can be categorized as 'hot', as per the categorization of [15]. There is generally a reason for the popularity of these 'hot' queries, such as a sudden media focus or a general interest due to a recurring event (e.g., Valentine's Day).

Recognizing 'hot' queries programmatically can be performed by analyzing the temporal aspects of a query. If we can detect a peak in a query's submission frequency at the moment the peak occurs, a suggestion algorithm that takes these temporal aspects into account can probably generate more relevant suggestions than the generic algorithm. For example, if a user submits the query "china" soon after the occurrence of an earthquake, the suggestion "earthquake in china" is likely to be more relevant for the user than a more generic suggestion like 'china travel'.

Within our research we chose to use a simple metric to define whether or not a query is currently 'hot'. We analyze a given query's daily submission counts over the past 40 days from ilse.nl's query log and calculate the average and the standard deviation. If the difference between the average and the current peak is higher than three standard deviations, we define the query as 'hot'. As a pre-condition we require the peak to have a height of at least 50 submissions. In our pilot study this method performed accurately, as all as 'hot' identified queries were clearly experiencing a peak, and most queries could be related to current events. An example of the temporal course of a season-related query can be seen in figure 4.

To generate suggestions for 'hot' queries, we chose to make a variation on the generic suggestion algorithm that does not make use of the data from the query logs of the past 28 weeks, but only uses data from the past seven days. Via this data reduction the current peak is expected to have the biggest influence on the generated search suggestions, which leads to search suggestions that are in sync with the recent happenings causing the peak. The pilot study revealed that this algorithm indeed generated suggestions that are specific to current events.

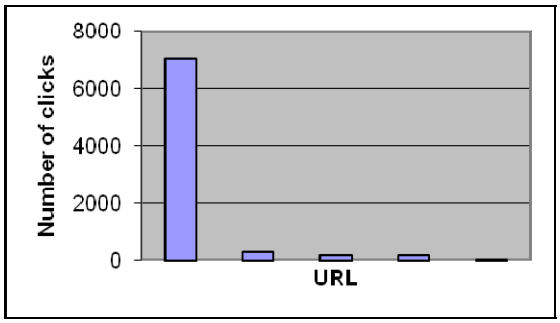


Figure 3. URL-click-distribution histogram for the navigational query "ing". Ninety percent of all clicks go to one URL.

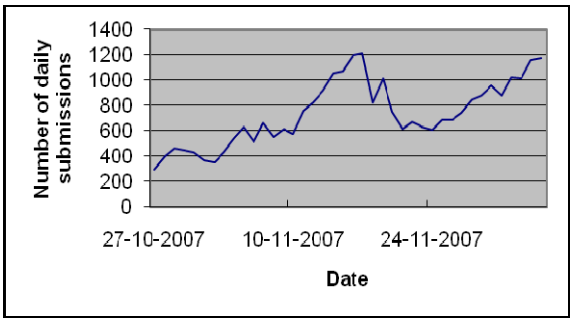


Figure 4. Daily query submissions for the season-related query "sinterklaas" (St. Nicolas, 5 December).

3.2.3 Goal-specific algorithm

By specifying the methods of recognition as well as generating goal-specific search suggestions, we can now specify the methods of the entire goal-specific algorithm: for each query we first check if the query is 'hot'. If so, the suggestions of the 'hot' algorithm are returned. If not, we check if the query is navigational. If that's the case, navigational search suggestions are returned. If the query is neither hot nor navigational, the query is considered to have 'another' goal, and the generic algorithm is used to generate suggestions.

4 Evaluation

4.1 Experimental setup

To test the effects of goal-specific search suggestions, both the generic and the goal-specific algorithms were implemented on the Dutch search engine ilse.nl. It should be noted that the available display space for the suggestions on the results pages is limited: by default no more than five suggestions are displayed, but if the resulting suggestions are lengthy, this number can be lower.

Search engine ilse.nl offers the possibility of A/B-testing, which allowed us to create two experiment conditions: in one condition the generic algorithm generated the search suggestions and in the other the goal-specific algorithm did. Users were randomly assigned to one of the conditions, and stayed in that condition for the duration of their search session.

The algorithms were implemented on December 21st 2007 and stayed online until January 1st 2008. During these twelve days all user behaviour – including search, display and click data – was logged. Requests that were likely to have come from automated robots were removed from the query logs.

To measure the effects of the suggestions of the algorithms, we used the following two metrics:

1. Perceived relevance of the suggestions. This was operationalized by measuring the click-through rate (CTR): the amount of clicks on search suggestions divided by the amount of displays of the search suggestions.
2. Effectiveness of the suggestions. This was operationalized by measuring the amount of clicks on search suggestions that were directly followed by a click on a search engine result. The rationale behind this metric is that if a user makes use of a search suggestion and subsequently clicks on one of the search engine results, this indicates that the suggestion helped the user find what he was looking for. On the other hand, if the user performed another action such as leaving the search engine or going back to the previous page, this was an indication that the suggestion did not help him to fulfill his information need.

4.2 Results

During the data collection period several million search queries were submitted to ilse.nl[2]. Table 3 shows the results of the generic and the goal-specific suggestion algorithms with regard to CTR and effectiveness. We expected the goal-specific algorithm to outperform the generic algorithm in both metrics. However, it shows that the generic algorithm has a higher CTR, while the goal-specific algorithm has a higher effectiveness. Both differences are significant (two-sample *z*-test for two population proportions, $\alpha < 0.0001$).

	Generic algorithm	**Goal-specific algorithm**
Click-through rate	1.9172 %	1.823 %
Effectiveness	54.148 %	56.74 %

Table 3. The click-through rate and effectiveness of the generic and goal-specific algorithm.

To be able to explain these differences we analyzed the performances of the goal-specific algorithm for each of the user goals (navigational and 'hot'). We calculated these by extracting a list of all submitted queries and their display and click counts. For each user goal we then posed the question: "If these queries would have been handled by the generic algorithm, what CTR and effectiveness would have been measured?" The results are listed in table 4.

	Generic algorithm		**Goal-specific algorithm**	
	Navigational queries	**'Hot' queries**	**Navigational queries**	**'Hot' queries**
Click-through rate	1.785 %	3.70 %	1.410 %	4.90 %
Effectiveness	54.67 %	52.7 %	55.11 %	61.8 %

Table 4. The click-through rate and effectiveness of the generic and goal-specific algorithm, split up per user goal.

For navigational queries, the CTR is significantly higher for the search suggestions generated by the generic algorithm than by the goal-specific algorithm ($\alpha < 0.0001$). The difference in effectiveness is not significant. The difference in CTR also turns out to be the cause of the CTR-performance of the goal-specific algorithm as a whole, as the amount of as navigational identified queries within our evaluation was around 30 %.

It's not surprising that navigational queries generally have a lower CTR than other queries. After all, the user's intent is to reach a web site or page directly, thus not paying much attention to the available search suggestions. However, the fact that the suggestions of the generic algorithm were perceived as more relevant than those of the goal-specific algorithm is surprising. There are several possible explanations. The first two are the identification precision and quality of suggestions. However, our pilot study indicated that there were no issues with either of these. The third explanation can be found in the limited amount of display space on the results page. As described earlier, navigational search suggestions are preceded by the original query if necessary. This results in lengthier suggestions, thus limiting the number of suggestions that can actually be displayed on the results page. This suspicion is confirmed after examination of the query logs. Less suggestions were displayed for navigational queries by the goal-specific algorithm ($\mu = 3.1$ and $\sigma = 1.1$) than by the generic algorithm ($\mu = 4.5$ and $\sigma = 0.8$). This may have resulted in a lower CTR. A final explanation can be found at a user level. The average user with a navigational goal generally ignores the available search suggestions. However, users with a non-navigational goal, whose query is incorrectly classified as navigational, will be more likely to make use of the suggestions. For those users, the domain-specific suggestions will be less relevant than generic suggestions. However, to confirm whether this is in fact a decisive factor on the results we found more research is required.

For 'hot' queries, the CTR of the goal-specific algorithm is significantly higher than the generic algorithm ($\alpha < 0.0001$), as is the effectiveness ($\alpha < 0.001$). These results confirm our expectations, that identifying user goals can improve search suggestions. It is interesting to see that 'hot' queries are

[2] We are not at liberty to disclose the exact number of visitors to search engine ilse.nl.

characterized by a high CTR, regardless of which algorithm produced the suggestions. Apparently users who are looking for information on a recently popular subject are more inclined to make use of the search suggestions available. When these suggestions are in fact geared towards the most recent events (i.e. generated by the goal-specific algorithm), this increased the perceived relevance with more than 32%. It also turns out that these goal-specific suggestions actually contributed to more effective search sessions: the measured effectiveness increased by over 17%. In other words, goal-specific search suggestions clearly outperform generic search suggestions for this user goal. This proves that the application of methods for identifying user goals and generating more specific suggestions can improve a user's search experience.

5 Conclusions and discussion

In this work we presented an algorithm to recognize the goals of users of a search engine automatically. We found that the algorithm could reliably classify the queries of these users into three classes: queries aimed at finding one specific web site (navigational queries), queries aimed at finding information that has recently become popular ('hot' queries) and queries with another goal (other queries).

The classification was used to suggest alternative queries to search engine users. We extended the successful search suggestion algorithm of Beeferman and Berger [1], which generates alternative queries without taking the goals of the users into account. For navigational queries, we limited the algorithm of Beeferman and Berger to pages within the domain of the most relevant web site. For hot queries we ran the algorithm only on the most recent log data. These extensions make the suggestions more tailored to the needs of the current user, without limiting the applicability of the algorithm or increasing computational complexity.

Evaluation on a real life search engine clearly showed the potential of goal-specific search suggestions. For queries that were classified as 'hot', the suggestions generated by the goal-specific algorithm were used more often than the suggestions from the original algorithm of Beeferman and Berger. Moreover, the goal-specific suggestions more often led to successful search sessions. This shows that taking search goals into account can significantly improve search suggestions.

The goal-specific algorithm proved effective for 'hot' queries, but for navigational queries it was outperformed by the original algorithm. Apparently, restricting the suggestions to one domain does not help users with navigational goals. More research is needed to determine what type of search suggestions is suited for these users.

In this study the temporal aspect of queries was limited to two classes: queries that were 'hot' and queries that were not 'hot'. Suggestions for hot queries were based on 1 week of log data. For other queries 28 weeks of data were used. This distinction already led to large improvements, but possibly even better suggestions can be given if we determine exactly which part of the log data is still relevant.

Another direction for future research involves recognizing more detailed search goals. For instance, the meaning of the certain queries, such as 'upcoming events', depends not only on the time at which the query is issued, but also on the location of the user. We may be able to recognize this by analyzing the relation between the locations from which queries are issued and the URLs that are clicked.

Finally, determining the goals of search engine users can be useful for other purposes than the generation of search suggestions. Possibly, we can improve the precision of the search engine itself by placing the search results that match the user's goals at a higher position in the results list.

Acknowledgements

This research would not have been possible without the generous cooperation of the research department of Ilse Media. Special thanks go out to Jochem Prins for support and involvement, as well as Nils Rooijmans for offering this research opportunity.

References

[1] D. Beeferman and A. Berger. Agglomerative Clustering of a Search Engine Query Log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*:407–416, 2000.

[2] A. Broder. A Taxonomy of Web Search. *ACM SIGIR Forum* 36(2):3-10, 2002.

[3] ilse.nl, http://ilse.nl/

[4] B.J. Jansen, A. Spink and T. Saracevic. Real Life, real Users, and real Needs: a Study and Analysis of User Queries on the Web. *Information Processing and Management*, 36(2):207-227, 2000.

[5] D. Johnson, V. Malhotra and P. Vamplew. More Effective Web Search Using Bigrams and Trigrams. *Webology*, 3(4), 2006.

[6] I.-H. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*:64-71, 2003.

[7] D. Kelly, V.D. Dollu and X. Fu. The Loquacious User: A Document-Independent Source of Terms for Query Expansion. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*:457-464, 2005.

[8] R. Kraft and J. Zien. Mining Anchor Text for Query Refinement. In *Proceedings of the 13th international conference on World Wide Web*:666-674, 2004.

[9] U. Lee, Z. Liu and J. Cho. Automatic Identification of User Goals in Web Search. In *Proceedings of the 14th international conference on World Wide Web*:391-400, 2005.

[10] E. Meij and M. de Rijke. Thesaurus-Based Feedback to Support Mixed Search and Browsing Environments. In *11th European Conference on Research and Advanced Technology for Digital Libraries*, 2007.

[11] B.V. Nguyen and M.-Y. Kan. Functional faceted Web Query Analysis. In *Proceedings of WWW 2007 Workshop on Query Log Analysis*, 2007.

[12] S.Y. Rieh and H. Xie. Patterns and Sequences of Multiple Query Reformulations in Web Searching: a Preliminary Study. In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology* 38:246-255, 2001.

[13] D.E. Rose and D. Levinson. Understanding User Goals in Web search. In *Proceedings of the 13th international conference on World Wide Web*:13-19, 2004.

[14] C. Silverstein, M. Henzinger, H. Marais and M. Moricz. Analysis of a very large Web Search Engine Query Log. *ACM SIGIR Forum*, 33(1):6-12, 1999.

[15] J. Zien, J. Meyer and J. Tomlin. Web Query Characteristics and their Implications on Search Engines. IBM Research Report, RJ 10199, 2000.

Deep Belief Networks for Dimensionality Reduction

Athanasios K. Noulas [a] Ben J.A. Kröse [a]

[a] *University of Amsterdam, 1098 SJ Amsterdam, The Netherlands*

Abstract

Deep Belief Networks are probabilistic generative models which are composed by multiple layers of latent stochastic variables. The top two layers have symmetric undirected connections, while the lower layers receive directed top-down connections from the layer above. The current state-of-the-art training method for DBNs is contrastive divergence, an efficient learning technique that can approximate and follow the gradient of the data likelihood with respect to the model parameters. In this work we explore the quality of the non-linear dimensionality reduction achieved through a DBN on face images. We compare the results achieved to the well know Principal Component Analysis as well as with a Harmonium model, which is the top layer of a DBN.

1 Introduction

Dimensionality reduction in statistics refers to the process of reducing the number of random variables at hand. In machine learning, these random variables correspond usually to features of our domain, and the process of reducing their number can be seen as feature selection or extraction. Dimensionality reduction has been the subject of numerous studies, since real-world problems are very high-dimensional and reducing the number of dimensions is essential to boost the efficiency of machine learning algorithms.

There are many different criteria on the basis of which one can reduce the dimensionality of a dataset. A very common technique is Principal Component Analysis (PCA), where we select the linear projections of the data which will result in maximum variance, in the hope that the lower dimensionality space will produce easily separable classes for classification. Dimensionality reduction can be represented as a neural network of multiple layers, visible in Figure 1, which is called autoencoder or autoassociator. The network is trained to discover a lower representation of the data, lying in the middle layer, that will allow optimal reconstruction. If there is only one linear hidden layer with k nodes, the autoencoder will project the data in the span of the k first principal components of the dataset, thus becoming identical to PCA. However, if the hidden layer is non-linear, different kinds of multi-modal abstraction with possibly better results are feasible [3].

In modern artificial intelligence, the most popular framework is undoubtedly probabilistic models. The problem of dimensionality reduction can be seen as a two layer model: the bottom layer compromises of observable variables and corresponds to the input vector, while the top layer compromises from hidden variables and corresponds to the reduced-dimensionality space. The more recent probabilistic PCA and multinomial PCA can be seen as a realisation of such a directed graphical model. The undirected version of such two layer models was first introduced in [5] and is called harmonium. In this work we consider Harmoniums with multinomial visible variables and continuous, Gaussian hidden variables, introduced in [6], as well as the Restricted Boltzmann Machines (RBMs) which have binary hidden and visible nodes [1]. Harmoniums and RBMs compromise the building blocks of a DBN [2]. We discuss the advantages and disadvantages of directed and undirected models in section 4.

A DBN is a multiple layer generative model. Roughly speaking, the bottom layer is observable, and the multiple hidden layers are created by stacking multiple RBMs on top of each other. The final layer is a Harmonium with Gaussian hidden nodes, which in our case correspond to the reduced dimensionality. We discuss briefly the theoretical advantages of deep architectures in section 4, but note here that in general there is no straightforward learning technique. The parameters are learnt approximately using Contrastive

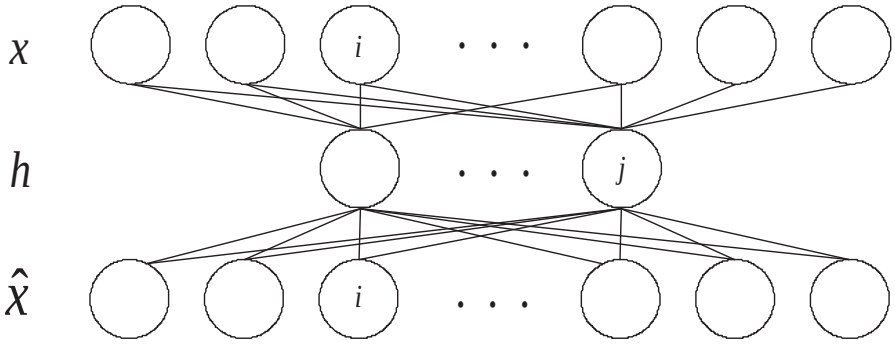


Figure 1: Graphical representation of an autoencoder. The input is the same as the output during training, while the middle layer corresponds to the lower dimensional representation.

Divergence learning [1], and the final deep architecture is fine-tuned based on a problem-specific objective function.

This paper is organised as follows. In section 2 we describe the details of the models used to perform dimensionality reduction. In section 3 we describe the dataset used and the experimental results acquired from these models. We conclude this paper with section 4 where not only the experimental results but also theoretical perspectives of the proposed models are discussed.

2 Model Specifications

In this section we describe the models evaluated on dimensionality reduction on our dataset. We start in 2.1 with a description of RBMs which is the simplest two layered model we may have. Their discrete nature makes them an ideal platform to present contrastive divergence learning which is described in section 2.2. In section 2.3 we describe the structure and application of contrastive learning in a harmonium with continuous Gaussian hidden variables. Finally in section 2.4 we present the DBN framework, learning, and fine tuning for dimensionality reduction.

2.1 Restricted Boltzmann Machines

A RBM is an energy-based model, which means that the probability distribution over the variables of interest is defined through an energy function. It is composed from a set of observable variables $x = \{x(i)\}$ and a set of hidden variables $h = \{h(i)\}$, as we can see in figure 2. The energy of a given configuration is estimated as:

$$Energy\,(x, h) = -b^{\top} x - x^{\top} h - h^{\top} W x \tag{1}$$

while the probability distribution over the configuration of the variables is

$$P\,(x, h) \propto e^{-Energy(x,h)} \tag{2}$$

note that we did not write the normalisation term $Z = \Sigma_{x,h} e^{-Energy(x,h)}$ in order to express our inability to compute it in general. The parameters W, b and h of the Energy function are learnt using a problem specific criteria. In the case of dimensionality reduction we would like to set them to those values that will allow optimal reconstruction of the input vector given it's low-dimensional representation.

In order to see how we can use an RBM for dimensionality reduction and reconstruction, consider an RBM with fewer nodes at the hidden layer than in the visible one. An important property of the RBM is that there are no connections among nodes of the same layer. Thus, we sampling from the conditional distributions $p\,(h|x)$ and $p\,(x|h)$, which factorise as:

$$P\,(x|h) = \prod_i P\,(x(i)|h) \tag{3}$$

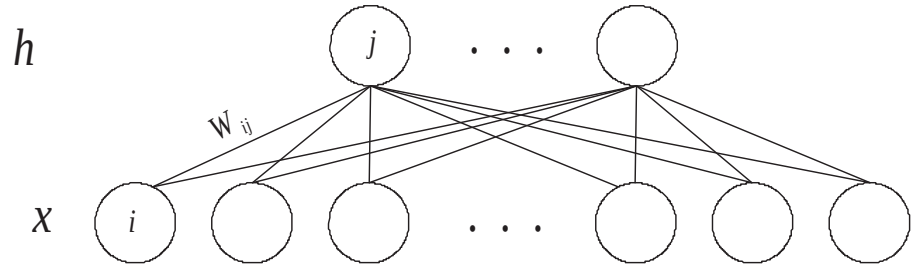


Figure 2: A Restricted Boltzaman machine. Each node of the hidden layer is connected to all the nodes of the visible layer, while there are no connections among nodes of the same layer.

$$P\left(h|x\right) = \prod_{j} P\left(h(j)|x\right) \tag{4}$$

Therefore, given an observation vector x we can sample a lower-dimension representation h. We can then use this representation to reconstruct $\hat{x}$.

In the case of the RBM, all the nodes are binary and the individual node probabilities are given as:

$$P\left(x(i)|h\right) = sigm(b_i + W_{\cdot i} \cdot h) \tag{5}$$

$$P\left(h(j)|x\right) = sigm(c_j + W_{j\cdot} \cdot x) \tag{6}$$

where $W_{\cdot i}$ and $W_{j\cdot}$ refer to the corresponding vectors and rows of matrix W.

In the learning face, we want to discover the parameters that maximise the probability of the vector x given it's encoding in the low dimensional space. That is maximise $p(x|\hat{h})$ with $\hat{h} \sim p(h|x)$, which corresponds to setting the parameters of the RBM to the values that maximise the data likelihood. We can rewrite the data likelihood as:

$$\begin{aligned} p\left(x,h\right) &= \frac{e^{-Energy(x,h)}}{Z} \\ p\left(x\right) &= \sum_{h} \frac{e^{-Energy(x,h)}}{Z} \\ &= \frac{e^{-FreeEnergy(x)}}{\sum_x e^{-FreeEnergy(x)}} \end{aligned}$$

where $FreeEnergy\left(x\right) = -\log \sum_h e^{-Energy(x,h)}$, with the term *Free Energy* coming from physics.

We can now maximise the data likelihood by gradient ascent, since the average log-likelihood gradient with respect to the parameters $\theta = \{W, b, h\}$ is:

$$E_{\hat{P}} \frac{\partial \log p\left(x\right)}{\partial \theta} = -E_{\hat{P}} \frac{\partial FreeEnergy(x)}{\partial \theta} + E_P \frac{\partial FreeEnergy(x)}{\partial \theta} \tag{7}$$

where $\hat{P}$ is the training set empirical distribution and P the distribution of our model. E_P denotes expected value under distribution P. We refer the interested reader to [1] for formal proof of the gradient formula.

2.2 Contrastive Divergence

Even with equation 7 at hand, it is not easy to train an RBM. As we can see in figure 3, we need to run a sampling algorithm multiple times for each training example in order to get a reconstruction useful to estimate the model's distribution. If sampling is repeated for a sufficient number of iterations over all the data set points, we can acquire the value of the gradient for the specific model parameters. Following the gradient in this manner is extremely expensive computationally, and in multi-dimensional problems practically unfeasible.

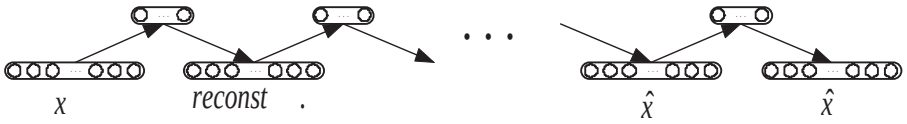


Figure 3: A sampling chain from the original vector x up to convergence to $\hat{x}$. The first reconstruction ($reconst.$) is used for contrastive divergence learning.

Instead, we can use contrastive divergence, a technique which has given very promising results. The technique is based on two approximations. The first approximation is replace the average over all possible inputs, as seen in equation 7, with a single sample. We update the parameters very often, after one or a few samples, and therefore indirectly we introduce some averaging which we expect to be sufficient. The second and most important approximation is to run the sampler for a single iteration, instead until the chain converges. In this way we expect the parameters to learn the values of the parameters that produce the minimum reconstruction error. From a macroscopic point of view, if the data distribution remains intact for a singe reconstruction, it will remain intact for the rest of the iterations, and thus we have converged to the final distribution from the first reconstruction. Once more we refer the interested reader to [1] for a broader discussion.

2.3 Harmonium

The harmonium is a RBM with continuous hidden nodes. Welling et. al in [6] introduced a harmonium with multinomial visible nodes, which proved to be extremely efficient in latent semantic indexing, and explored the theoretical possibilities and restrictions of this structure. It can also be trained using contrastive divergence, with $P\left(x(i)|h\right) = sigm(b_i + W_{.i} \cdot h)$ and $P\left(h(j)|x\right) = \mathcal{N}(c_j + W_{j.} \cdot x, 1)$ where $\mathcal{N}(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ. In case x is multinomial, we keep a separate weight for each possible $x(i)$, and we sample $x(i)$ from a soft-max function over the possible states [6].

2.4 Deep Belief Networks

A DBN with l layers, models the joint distribution of the observable layer x and the hidden layers h^k for $k = 1 : l$ as

$$p\left(x, h^1 \ldots, h^l\right) = p\left(x|h^1\right) \prod_{k=1:l-2} p\left(h^k|h^{k+1}\right) p\left(h^{l-1}, h^l\right) \tag{8}$$

where each of the conditional probabilities is modelled as an RBM, while the probability over the two top layers is modelled as a harmonium. A graphical representation of a DBN is visible in figure 2.4.

Training a DBN has two phases. In the first phase, we start by training the RBM of layers x and h^1. We keep adding consecutive layers treating the reconstructions acquired in the previous layer as data of the visible layer. For instance, we get the projection at h^1 and train an RBM between h^1 and h^2 using the reconstructions of h^1 as data. We train the top two layers model as an harmonium, using the reconstruction of h^{l-1} as data, and hidden Gaussian nodes in h^l. The first phase is performed using contrastive divergence, and we expect to get the model parameters near a good local maximum of the data likelihood function. In the second phase, we tune the parameters of the whole DBN based on a problem specific criteria. In the case of dimensionality reduction this is performed with back propagation, exactly the way it is performed for neural networks. The hope is that this phase will tune the parameters to the local maximum approached by the first phase.

3 Experiments

In this section we present the dimensionality reduction experiments conducted for this work. In section 3.1 we describe briefly the dataset we used, and in section 3.2 the objective and setup of our experiments. The numerical results acquired can be found in section 3.3

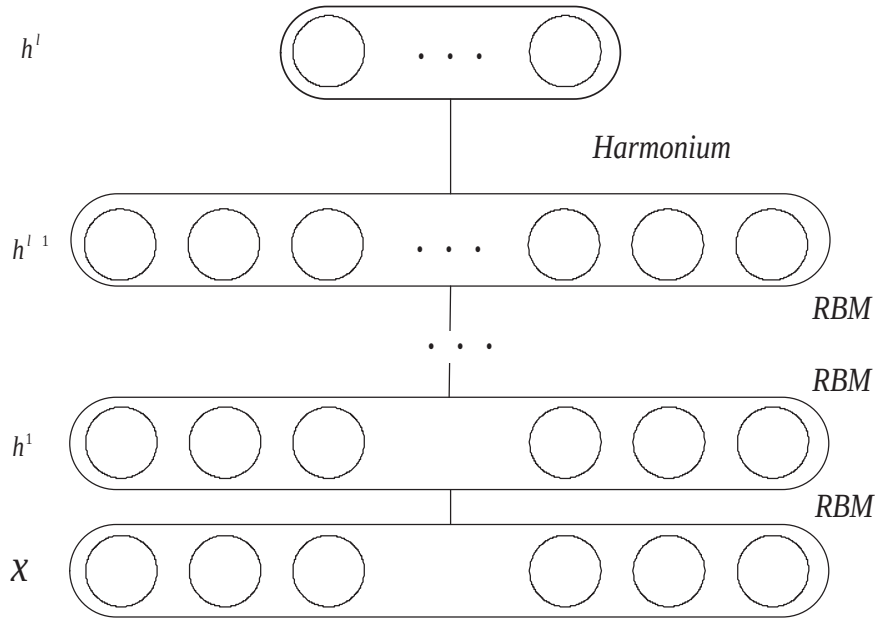


Figure 4: A Deep Belief Network with l layers of hidden nodes. The first $l-1$ layers are added as RBMs, using the projection on the last layer as data. The top two layers are a Harmonium with continuous Gaussian hidden variables, and corresponds to the reduced dimensional space.

3.1 Dataset

We used 400 faces coming from the AR Face Database [4]. The face images are 36×48 pixels, creating 1728 dimensional datapoints. The original grey valued images were turned into they binary counterparts in order to be usable as input for the RBM.

3.2 Experimental Setup

We extracted the principal components of the dataset and performed projection and reconstruction for a grid of 40 values between 2 and 300 dimensions. The same applied for harmoniums, where we trained different harmoniums for each lower space dimensionality. Finally, we trained different DBNs with 4 hidden layers. The first three hidden layers had 1000, 500 and 250 nodes, and the top layer took values in the low dimensional space. We evaluated each reconstruction using the sum of squared differences between the original value and the reconstructed one.

3.3 Results

In figure 6, we can see the reconstruction error for different dimensionality reductions. The experiments grid had the values $2:20$, $25:100$ and $110:300$. Furthermore, in figure 6, we can see how the reconstructed face images look like. We see that the harmonium outperforms PCA, while the DBN produces the best results for low number of retained dimensions.

This was expected for two reasons. Firstly, the Harmonium and the DBN perform non-linear dimensionality reduction which proves more efficient for this kind of data. Secondly, they are trained with main objective the optimal reconstruction of a vector from the hidden space, while PCA's main objective is to maximise the variance of the data in the lower dimensional space. The DBN outperforms the Harmonium when a few dimensions are retained because it learns the data distribution as a product of multiple layers. In this way, we can learn much more varying distributions. On the other hand, as more flexibility is available because we keep a large number of dimensions, the Harmonium uses it optimally while the DBN is trapped in a sub-optimal local minima. The image vectors require sharp distributions in order to discriminate face pixels from the background, and that's why this flexibility produces better results.

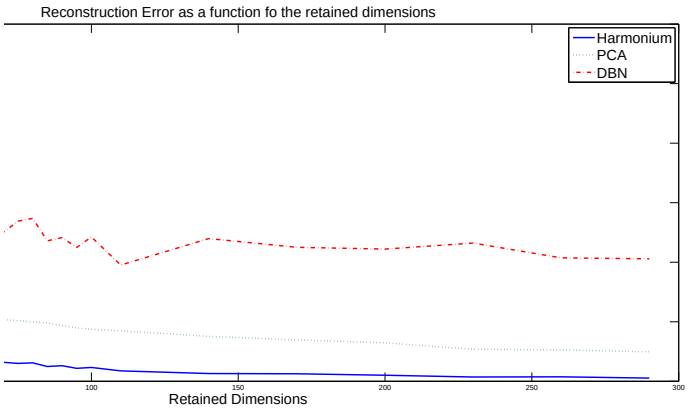


Figure 5: Sum of the squared error between the reconstructions and the original vectors as a function of the retained dimensions

4 Discussion & Conclusions

In this paper, an undirected deep architecture provided optimal results in dimensionality reduction. This comes in favour of some theoretical considerations that support the use of similar architectures for learning in artificial intelligence. The arguments regard the flexibility these architectures provide in terms of inference, mapping in the latent space and representational power.

Deep architectures have been avoided in the past because they are much harder to train. More specifically, if one tries to train directly a deep architecture, the two top layers tend to get meaningful, criteria-related weights, while the rest just perform useless perturbation of the data. Training with contrastive divergence overcomes this difficulty. In the first phase we create multiple perturbations of the data, retaining the information contained in it. Thus, we hope that when optimising our problem-specific objective, we will be able to utilise the layer containing the optimal data reconstruction. The question that arises naturally is how many layers do we need for a given problem, and it cannot be answered directly. However, we should keep in mind that a DBN can use large quantities of unlabelled data to learn the structure of a given dataset. We can then use a few labelled examples to perform high accuracy classification in this well structured space.

This view is very similar to human learning. Babies observe the world around them in an unsupervised manner, and afterwards they use a few labelled examples to perform classification in the space they have structured. For example, someone can use thousands of face images in an unsupervised way, and then based on the task at hand use labelled examples of male or female subjects, to classify genders or different ages to perform regression in the age space.

Another important quality of undirected graphs in comparison to the directed ones is that we can sample the configuration of the hidden nodes directly, since they are independent of each other given the visible ones. This can be very useful in applications that require fast lower-space representations like for instance document retrieval, with an example of a harmonium performing document retrieval and discovering word structure given in [6]. Finally, we can use the trained DBN to generate data from our training distribution.

References

[1] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.

[2] Geoffrey E. Hinton. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[3] Nathalie Japkowicz, Stephen Jose Hanson, and Mark A. Gluck. Nonlinear autoassociation is not equivalent to PCA. *Neural Computation*, 12(3):531–545, 2000.

[4] A.M. Martinez and R. Benavente. The ar face database. *CVC Technical Report 24, http://rvl1.ecn.purdue.edu/ aleix/aleixfaceDB.html*, June 1998.

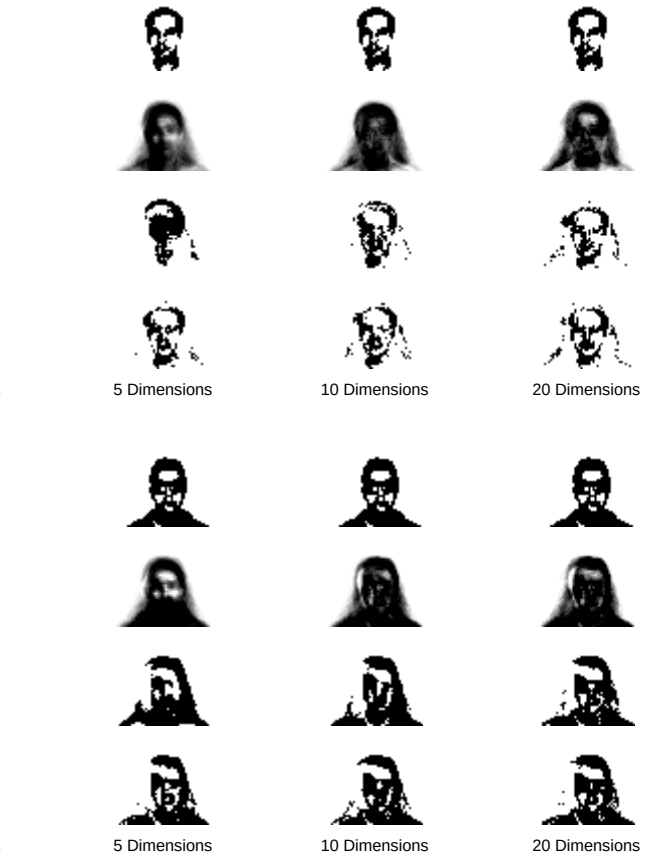


Figure 6: From top-down we have different datapoint visualizations for PCA, Harmonium and DBN. Their reconstructions from the different models for 3, 5, 10 and 20 retained dimensions from left to right

[5] P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pages 194–281, 1986.

[6] Max Welling, Michal Rosen-Zvi, and Geoffrey Hinton. Exponential family harmoniums with an application to information retrieval. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1481–1488. MIT Press, Cambridge, MA, 2005.

Human Gesture Recognition using Sparse B-spline Polynomial Representations

A. Oikonomopoulos [a] M. Pantic [a] I. Patras [b]

[a] *Computing Department, Imperial College London*
[b] *Electronic Engineering Department, Queen Mary University of London*

Abstract

The extraction and quantization of local image and video descriptors for the subsequent creation of visual codebooks is a technique that has proved extremely effective for image and video retrieval applications. In this paper we build on this concept and extract a new set of visual descriptors that are derived from spatiotemporal salient points detected on given image sequences and provide local space-time description of the visual activity. The proposed descriptors are based on the geometrical properties of three-dimensional piecewise polynomials, namely B-splines, that are fitted on the spatiotemporal locations of the salient points that are engulfed within a given spatiotemporal neighborhood. Our descriptors are inherently translation invariant, while the use of the scales of the salient points for the definition of the neighborhood dimensions ensures space-time scaling invariance. Subsequently, a clustering algorithm is used in order to cluster our descriptors across the whole dataset and create a codebook of visual verbs, where each verb corresponds to a cluster center. We use the resulting codebook in a 'bag of verbs' approach in order to recover the pose and short-term motion of subjects at a short set of successive frames, and we use Dynamic Time Warping (DTW) in order to align the sequences in our dataset and structure in time the recovered poses. We define a kernel based on the similarity measure provided by the DTW in order to classify our examples in a Relevance Vector Machine classification scheme. We present results from two different databases of human actions that verify the effectiveness of our method.

1 Introduction

Vision-based analysis of human motion is nowadays one of the most active fields of computer vision, due to its practical importance for a wide range of vision-related applications, like video retrieval, surveillance, vision-based interfaces and Human-Computer Interaction. From any given video sequence humans are usually able to deduce information about its content quickly and easily. When it comes to computers, however, robust action recognition still remains a very challenging task, evident from the abundance of different motion analysis approaches that have been developed [13].

Typically, activity recognition systems can be divided into two main categories. The first concerns tracking of body parts and the subsequent use of the resulting trajectories for recognition [17], [5]. These approaches, however, are highly dependent on the type of tracking system that is used and its target application. In addition, due to the deformable nature and articulated structure of the human body, these methods suffer from problems like accurate initialization, occlusion and high dimensionality.

A second category of systems uses sets of spatiotemporal feature descriptors in order to represent human body motion. The concept of spatiotemporal feature extraction for activity recognition stems from the domain of object recognition, where static features have been successfully used for the detection of various objects from images [8], [1]. In [7], a Harris corner detector is extended in the temporal domain, leading to a number of corner points in time, called space-time interest points. The resulting interesting points correspond roughly to points in space-time where the motion abruptly changes direction. In [3], human actions are treated as three-dimensional shapes in the space-time volume. The method utilizes properties of the solution to the Poisson equation to extract space-time features of the moving human body, such as local space-time saliency, action dynamics, shape structure and orientation. In [14] a local self-similarity descriptor is extracted in order to match areas in images or videos that share similar geometric properties.

Finally, in [6] a set of spatiotemporal features inspired from the human visual cortex, called C features, are extracted for the recognition of human and animal motions. The method works in an hierarchical way and the obtained features are invariant to scale changes in space and time.

Recently a number of works used visual codebooks in order to detect and recognize objects and/or humans. The visual codebook creation is performed by grouping the extracted feature descriptors in the training set using, for instance, a clustering algorithm [10]. The resulting centers are then considered to be codewords and the whole set of codewords forms a 'codebook'. In a 'bag of words' approach each instance is represented as a histogram of codewords, and recognition is performed by histogram comparison. In [2] a set of SIFT-like features are hierarchically used in order to form 'hyperfeatures' for the purpose of object recognition, while in [4] static and dynamic features based respectively on gradients and optical flow are extracted in order to detect humans in image sequences. In order to further enhance the performance of these models, several researchers have gone one step forward and encoded the spatial relationships that exist between the features. In [9], extracted features cast votes towards the center of the object from which they are extracted. In this way the system implicitly encodes the spatial relationships between the extracted features. In [15] a similar enhancement takes place by considering pairs of visual words which co-occur within local spatial neighborhoods, denoted as 'doublets'. In [11] constellations of static and dynamic bags of features are modeled in order to recognize human activities. Finally in [16], SIFT descriptors are extracted from spatial video patches and their spatial layout is encoded for the purpose of video or image retrieval.

In this paper we extract a new set of visual descriptors that are derived from the spatiotemporal salient points of [12]. At each salient point location we define a spatiotemporal neighborhood with dimensions proportional to the detected space-time scale of the point. We use the locations of the salient points that are engulfed within this neighborhood in order to approximate a three dimensional piecewise polynomial, namely a B-spline. Our descriptors are subsequently derived from the geometrical properties of each polynomial as these are captured in their partial derivatives of different orders. At the next step, the whole set of descriptors is accumulated into a number of histograms, depending on the number of parameters that describe the spline and the maximum degree of its derivatives. Since our descriptors correspond to geometrical properties of the spline, they are translation invariant. Furthermore, the use of the automatically detected space-time scales of the salient points for the definition of the neighborhood ensures invariance in space and time. Similar to other approaches, where a codebook of visual words is created from appearance descriptors, we create a codebook of visual verbs by clustering our motion descriptors across the whole dataset. We use the resulting codebook in a 'bag of verbs' approach in order to recover the pose and instantaneous motion of subjects at a short set of successive frames and we use a Dynamic Time Warping scheme (DTW) in order to structure in time the recovered poses. We use the similarity measure between the examples, provided by the DTW, in order to define a kernel for a classifier based on Relevance Vector Machines (RVM). We present results in two different databases of human actions that verify the effectiveness of our method. Finally, we perform experiments in order to verify the generality of our descriptors, that is, their ability to encode and discriminate between unseen classes.

One of the main contributions of the method presented in this paper is the sparsity of the extracted descriptors, since they are extracted at spatiotemporal regions that are detected at sparse locations within the image sequence. This is contrary to the work of Blank et al [3], where a whole image sequence is represented as a space-time shape. Furthermore, the use of DTW adds structure to the recovered short-term motions of the subjects, as opposed to [3], [6], where features are matched based on the maximum similarity accross a whole video sequence. Our results are comparable [3], [6] or show improvement [11] with state of the art methodologies for the same sequences.

The remainder of the paper is organized as follows: in section 2 we describe our feature extraction process, including our B-spline approximation and the creation of our codebook. In section 3 we present our recognition method, that includes the DTW and RVM steps. In section 4 we present our experimental results and finally, in section 5 our final conclusions are drawn.

2 Representation

In this section we introduce the visual descriptors that we use in order to represent an image sequence. We will initially give some basics on B-splines and we will subsequently describe their usage in extracting local spatiotemporal image sequence descriptors. Finally, we will briefly explain the process that we followed in order to create a codebook from these descriptors.

2.1 B-spline Surfaces

Let us define an $M \times N$ grid of control points $\{P_{ij}\}$, $i = 1 \dots M$ and $j = 1 \dots N$. Let us also define a knot vector of h knots in the u direction , $U = \{u_1, u_2, \dots, u_h\}$ and a knot vector of k knots in the v direction, $V = \{v_1, v_2, \dots, v_k\}$. A B-spline surface of degrees p,q in the u and v directions respectively is given by:

$$F(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) P_{ij}, \tag{1}$$

where $N_{i,p}(u)$ and $N_{j,q}(v)$ are B-spline basis functions of degree p and q, respectively, defined as:

$$\begin{aligned} N_{i,0}(u) = \ \{ \quad & \begin{matrix} 1, & if\ u_i < u < u_{i+1}\ and\ u_i < u_{i+1} \\ 0, & otherwise \end{matrix} \\ N_{i,p}(u) = & \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{2}$$

The set of control points is referred to as the control net, while the range of the knots is usually $[0, 1]$. For this work we assume 3^{rd} degree polynomials, that is, $p = q = 3$.

2.2 Spatiotemporal Descriptors

In order to approximate a B-spline polynomial we need to initially define its control net, that is, P_{ij}. Formally, for each salient point location we want to approximate a polynomial having as control net the points within a small neighborhood O around the point in question. For a good approximation, however, ordering of the control points in terms of their spatiotemporal location is an important factor in order to avoid loops. In order to make this more clear, let us consider a set of points $L = \{l_i\}$ sampled uniformly from a circular curve. In order for a polynomial to approximate the circular curve from the L points, these points should be given in sequence, that is, $L = \{l_1, l_2, \dots, l_n\}$. If this is not the case, then the polynomial will attempt to cross the points in a different order, creating unwanted loops. Furthermore, it is clear that any points enclosed by the circle will also degrade the approximation and should not be accounted for. In order to overcome these problems, we perform two preprocessing steps on the set S: In the first step, we eliminate points that are enclosed within the closed surface defined by the boundary. In our implementation, a point lies in the boundary if it lacks any neighbors within a circular slice shaped neighborhood of radius r, minimum angle a and having the point as origin. For our implementation we selected a radius of 10 pixels and an angle of 70 degrees. In the second step, we order the selected boundary points. We do this by randomly selecting a point on the boundary as a seed and by applying an iterative recursive procedure that matches the seed point with its nearest neighbor in terms of Euclidean distance. This process repeats itself having as seed the nearest neighbor selected until there are no nearest neighbors left, that is, either an edge has been reached or all points have been accessed. One could argue that the procedure described above would select points in the convex hull of the motion, creating problems in the case of non-stationary background or if there are more than one subjects performing activities in the same scene. This however, is not true, as the whole procedure is performed locally. In effect, the amount of locality is determined by the radius r.

Let us denote by $S' = \{(\vec{s}'_i, \vec{c}'_i, y'_{D,i})\}$ the set of spatiotemporal salient points located on the motion boundary, obtained from the procedure of the previous section. For each salient point position within S' we define a spatiotemporal neighborhood N of dimensions proportional to $\vec{s}'_i$. Let us denote by O' the set of points in N. Then, for each N, we approximate a B-spline polynomial as in eq. 1. The grid of control points P_{ij} in eq. 1 corresponds to the set O', that is, each P_{ij} is a point in space-time. We should note that the grid is not and does not need to be uniform, that is, the pairwise distances of the control points can be different. The knot vectors U and V are a parameterization of the B-spline, and essentially encode the way the B-spline surface changes with respect to its control points. More specifically, the knot vector U encodes the way the x coordinates change with respect to y, while the knot vector V encodes the way both x and y change with respect to time t. Using this process, any given image sequence is represented as a collection of B-spline surfaces, denoted as $\{F_i(u,v)\}$. The number of surfaces per sequence depends on the number of points in S', since we fit one surface per salient point position. An example of a spline fitted to a set of points is presented in Fig. 1. Each member of the set $\{F_i(u,v)\}$ is essentially a piecewise polynomial in a three dimensional space. This means that we can fully describe its characteristics by means of its partial derivatives with respect to its parameters u, v. That is, for a grid of knots of dimensions $k \times h$ we calculate the following matrix R_i of dimensions $(pq-1) \times (hk)$:

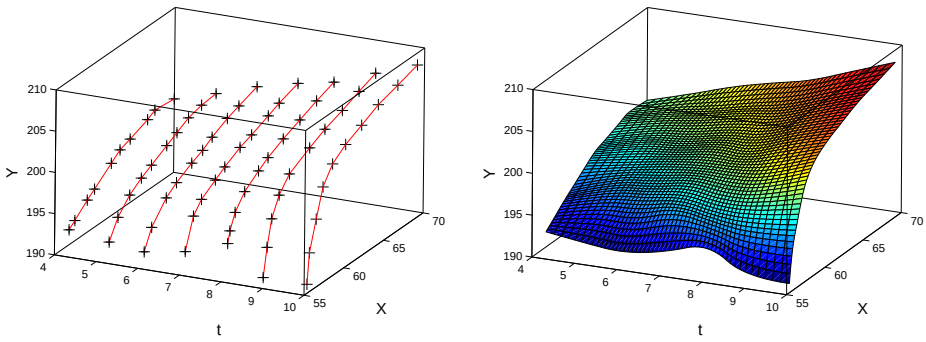


Figure 1: (a) A set of points within a spatiotemporal neighborhood N and (b) their B-spline approximation

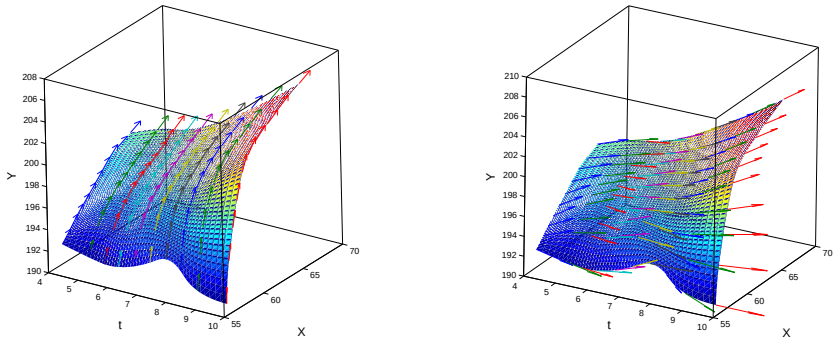


Figure 2: First derivatives with respect to (a) u and (b) v, drawn as three dimensional vectors

$$R_i = \begin{bmatrix} \frac{\partial F_i(u_1,v_1)}{\partial u} & \cdots & \frac{\partial F_i(u_h,v_k)}{\partial u} \\ \vdots & \ddots & \vdots \\ \frac{\partial^{(p-1)(q-1)} F_i(u_1,v_1)}{\partial u^{p-1}\partial v^{q-1}} & \cdots & \frac{\partial^{(p-1)(q-1)} F_i(u_h,v_k)}{\partial u^{p-1}\partial v^{q-1}} \end{bmatrix} \tag{3}$$

where $\partial^p/\partial u^p$ is the $p-th$ partial derivative with respect to u. From eq. 1 it is apparent that for specific values of u, v, $F_i(u, v)$ is a point in space-time, that is, a 3×1 vector. Consequently, each element of R_i is a vector of the same dimensions. In Fig. 2 an illustration of the first derivatives with respect to u and v is given. The derivatives are drawn as three dimensional vectors, superimposed on the spline from which they were extracted.

Our goal is to be able to represent each F_i with a single descriptor vector. For this reason, we bin each row of R_i into a single histogram of partial derivatives and we concatenate all the resulting $(pq-1)$ histograms into a single descriptor vector. This vector constitutes the descriptor of F_i and consequently the descriptor of a specific region in space and time of the image sequence. By repeating this process for each F_i, we end up with a set of descriptors for the whole sequence.

2.3 Codebook Creation

In order to create a codebook, applying a clustering algorithm to the whole set of descriptors is time and memory consuming. According to the authors of [9], the way a vocabulary is constructed has little impact to the final classification results. We therefore follow their approach and randomly subsample our descriptor set. Subsequently, we cluster our randomly selected features using K-means clustering. The resulting cluster centers are the codewords and the whole set of codewords constitutes the codebook. For this work we used a total number of 1000 clusters, as a compromise between representation accuracy and speed.

3 Classification

Having constructed our codebook, our goal is to be able to represent and classify any test image sequence to one of the available classes in our training set. A conventional application of a 'bag of verbs' approach would dictate that each image sequence in the dataset is represented as a histogram of visual codewords drawn from the codebook. Using the codebook in this way for our specific set of descriptors resulted in recognition rate of about 60%, using a 1-NN classifier based on the χ^2 distance between the histograms of the test and training sequences. We follow instead a different approach and use the codebook in order to recover the pose and instantaneous motion of the subjects performing the actions at a short set of successive frames. By doing this, we essentially encode each video as a collection of instantaneous motions.

As we will show in the experimental results section, even though pose recovery and subsequent classification using just a chamfer distance based nearest neighbor approach works quite well, this is not sufficient, as we would like to be able to add some structure and order in the instantaneous motions that are being recovered. A possible solution would be to use a temporal model like a Hidden Markov Model in order to encode the temporal relationships between the poses. This solution however is not practical, as the high dimensionality of the codebook would make the training of such a model cumbersome, especially in estimating the emission probabilities of the model. The use of a classification method that would be able to automatically provide these probabilities is not very practical either, as this would require manual annotation of similar poses between different examples of the same class. In order to deal with these issues, we decided to use Dynamic Time Warping (DTW) to align our sequences and subsequently apply a discriminant classifier like a Relevance Vector Machine (RVM) [18] for classification.

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is a well established technique for aligning any two sequences. The sequences are "warped" non-linearly in time in order to determine a measure of their similarity independent of certain non-linear variations in the time dimension. In order to use DTW for our problem, we consider as a sequence the series of the recovered instantaneous motions of each example, each being represented as a histogram of codewords. Since we are dealing with histograms, a suitable distance metric to use would be the χ^2 distance. Using this distance, we align our test sequences with every sequence in our training set. This procedure results in a similarity measure between the testing and training sequences, which is subsequently used in an RVM classification step.

3.2 Relevance Vector Machine

A Relevance Vector Machine Classifier (RVM) is a probabilistic sparse kernel model identical in functional form to the Support Vector Machine Classifier (SVM). Given a dataset of N input-target pairs $\{(F_n, l_n), 1 \leq n \leq N\}$, an RVM learns functional mappings of the form:

$$y(F) = \sum_{n=1}^{N} w_n K(F, F_n) + w_0, \tag{4}$$

where $\{w_n\}$ are the model weights and $K(.,.)$ is a Kernel function. For our work, we use the similarity measure provided by the DTW of the previous section in order to define a kernel for the RVM. More specifically, we apply the logistic sigmoid function to the DTW similarity measure in order to obtain a distance measure instead. Subsequently, we use a Gaussian RBF to define the kernel, that is,

$$K(F, F_n) = e^{-\frac{D(F,F_n)^2}{2\eta}}, \tag{5}$$

where D is the logistic sigmoid function of the DTW similarity measure and η is the width of the kernel. In the two class problem, a sample F is classified to the class $l \in [0, 1]$ that maximizes the conditional probability $p(l|F)$. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F)$,$1 \leq i \leq L$ is maximized:

$$Class(F) = \arg\max_i(p_i(l|F)). \tag{6}$$

Class	R/P (NN)	R/P (DTW)	R/P (RVM)	Confusion Matrix								
bend	1.0/1.0	0.88/1.0	1.0/0.9	**9**	0	0	0	0	0	0	1	0
jack	1.0/0.9	1.0/1.0	1.0/1.0	0	**9**	0	0	0	0	0	0	0
jump	0.67/0.67	0.56/1.0	0.78/0.88	0	0	**7**	0	0	1	0	0	0
pjump	0.89/1.0	1.0/1.0	1.0/1.0	0	0	0	**9**	0	0	0	0	0
run	1.0/0.71	1.0/0.56	1.0/0.9	0	0	1	0	**10**	0	0	0	0
side	0.89/1.0	0.89/1.0	0.78/1.0	0	0	0	0	0	**7**	0	0	0
walk	0.5/0.71	1.0/0.83	1.0/0.83	0	0	1	0	0	1	**10**	0	0
wave1	1.0/1.0	0.78/1.0	0.78/1.0	0	0	0	0	0	0	0	**7**	0
wave2	1.0/1.0	0.78/1.0	1.0/0.9	0	0	0	0	0	0	0	1	**9**
Total	0.88/0.88	0.89/0.93	0.93/0.93									

Table 1: Recall and Precision rates for the kNN and RVM classifiers on the Weizmann dataset

Class	R/P (SP-RVM)	R/P (NN)	R/P (DTW)	R/P (RVM)
1	1.0/1.0	0.9/1.0	1.0/1.0	1.0/1.0
2	1.0/0.63	1.0/0.91	1.0/0.83	1.0/0.83
3	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
4	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
5	0.7/0.78	1.0/0.56	0.4/0.67	1.0/0.91
6	0.6/0.5	0.3/0.3	0.6/0.35	0.8/0.8
7	0.7/1.0	0.9/1.0	1.0/0.91	0.9/0.9
8	1.0/1.0	1.0/1.0	0.9/1.0	0.9/0.9
9	1.0/1.0	1.0/0.67	1.0/0.71	0.9/0.9
10	0.7/0.78	0.1/0.5	0.4/0.67	0.8/0.89
11	1.0/1.0	1.0/1.0	1.0/1.0	0.9/1.0
12	1.0/1.0	1.0/0.91	1.0/1.0	1.0/1.0
13	0.7/1.0	0.9/1.0	0.8/1.0	0.8/1.0
14	1.0/1.0	1.0/0.91	1.0/1.0	1.0/0.91
15	1.0/1.0	0.5/1.0	0.7/1.0	1.0/1.0
Total	0.89/0.91	0.84/0.85	0.85/0.88	0.93/0.94

Table 2: Recall and Precision rates for the kNN and RVM classifiers on the aerobics dataset

4 Experimental Results

In order to evaluate the proposed method we use two different datasets. The first is the one used in [3], containing 9 different actions such as walking, running and jumping. The second dataset [1] is one created by our group, containing a set of 15 different aerobic exercises, performed twice by five different subjects.

We performed our experiments in the leave-one-subject-out manner. That is, in order to classify a test exercise performed by a specific test subject, we created a codebook and trained the respective classifiers using all available data except for those belonging to the same class and performed by the same subject as the test exercise. We present three different sets of classification results. In the first set, each frame of a test sequence is matched with the closest frame of a training sequence in terms of their χ^2 distance and an overall distance measure is calculated as the sum of the minimum calculated frame distances. The test example is then classified to the class of the training example with the smallest overall distance (Chamfer distance). In the second set, each test example is classified to the class of the training example with the highest similarity, as this is calculated by the DTW procedure. Finally, we present results using an RVM classifier according to eq. 6. In Table 1 we present our classification results for the Weizmann dataset, in the form of recall and precision rates. Similar classification results for the aerobics dataset are given in Table 2. In the same Table, we also show classification results on this dataset based on the algorithm of [12] (denoted as SP-RVM), in which only the location of the spatiotemporal points was considered . As we can see, there is considerable improvement, which demonstrates the descriptive power of the proposed B-spline based representation.

As we can see from Tables 1 and 2, there is a slight increase in classification performance in the Weizmann and aerobics datasets using DTW, while there is a considerable increase of almost 5% by additionally

[1] This dataset is available from the author's website

using RVM. Although the increase is small, the use of DTW adds structure and consistency to the representation. In general, introduction of structure is important and expected to show benefits in datasets with larger number of classes. Using DTW, frames that are far apart from each other in terms of time cannot be matched. In the case of a classification method with no temporal structure, these kind of restrictions do not exist, and a frame in the beginning of a sequence can be matched with any frame of another sequence, as long as their χ^2 distance is small.

The average recall rate for the Weizmann dataset is about 93%. From the confusion matrix of the Table 1, we notice that there are reasonable confusions between similar classes like *jump*, *run*, *walk* and *side*, as well as *wave1* and *wave2*. Concerning the results on the aerobics dataset, we notice from Table 2 that there is low performance on classes $5, 6$ and 10 for the NN and DTW classifications, which considerably increases using the RVM classifier. The reason for this is that these classes are very similar and concern motions like squatting with an upright torso or bending while the subject is facing the camera. In order to discriminate between these motions, depth information is important, and since our features stem from salient point representations, they have difficulty recovering it.

Compared to the work of [3] and [6], our classification results are almost 4% lower. The use of DTW from our system, however, introduces structure to the recovered short-term motions and classification is performed based on this structure. On the contrary, in [3], [6] features are matched based on maximum similarity across whole image sequences. In addition, our system uses a sparse representation as opposed to [3], where a whole image sequence is represented as a space-time shape. Sparse, local representations, are shown to be significantly better in dealing with clutter and occlusions for object detection and recognition in comparison to global representations. Similar observations are expected to hold in the problem of action recognition. A sparse and structured representation is used in [11], where a recognition rate of 72.8% is reported on the Weizmann dataset, by far inferior to the 93% achieved by our method.

We used a leave-one-subject-out approach in order to evaluate our method. This means that for any test example, the created codebook contains information about the class of this example, although from different same-class examples. We would like to determine, if our features are general enough to handle completely unknown classes, that is, given a codebook of verbs how well is this codebook able to discriminate classes that did not contribute at all to its creation. Our motivation for this experiment lies in the fact that our system is able to consistently recover short-term motion in small spatiotemporal regions. Therefore, given an unknown class that shares a number of similar such regions with several known classes, there should be some limited ability for good discrimination. We performed two different experiments. In the first experiment we created a codebook from 14 classes of the aerobics dataset, completely excluding class 3, which was kept out for testing. The result was 8 out of 10 instances of the test class correctly classified. In the second experiment, we created a codebook from the whole aerobics dataset and tested it on the Weizmann dataset. The classes between these two datasets are completely different, except for the class *jack* of the Weizmann dataset which is similar to class 1 of the aerobics dataset and classes *wave1* and *wave2* of the Weizmann dataset which look like classes 2 and 7 of the aerobics dataset. The average recall rate for this experiment was 67.5%, with the worst performing classes being *jump*, *run* and *walk*. This result is reasonable, as these classes do not seem to share common poses with the ones in the aerobics dataset. These results indicate that it might be possible to use the proposed descriptors for representing new classes of actions. We intend to investigate on the issue of the size of the action database and perform the same experiments with features that are currently the state of the art in the field, like the features of [3], [6] and [11].

5 Conclusions

In this paper we presented a feature based method for human activity recognition. The features that we extract stem from automatically detected salient points and contain static information concerning the moving body parts of the subjects as well as dynamic information concerning the activities. We used the extracted features in order to recover the pose and the short-term motion of the subject in a 'bag of verbs' approach. Our results show that our representation is able to recover the kind of motion performed in a variety of different cases. Furthermore, our preliminary experiments show that our system is able to generalize well and handle unknown classes, which do not contribute to the creation of the utilized codebook at all.

Our future directions include additional experiments in order to determine the robustness of the proposed method in more challenging scenarios, like in the presence of dynamic background or moving camera. Furthermore, we intend to implement different, more efficient methods for codebook creation.

Acknowledgments

This work is in part financially supported by the MAHNOB project funded by the European Research Council under the ERC Starting Grant agreement No. ERC-2007-StG-203143.

References

[1] A. Pinz A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision*, 2006.

[2] A. Agarwal and B. Triggs. Hyperfeatures multilevel local coding for visual recognition. *European Conference on Computer Vision*, 1:30–43, 2006.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Proc. IEEE Int. Conf. Computer Vision*, 2:1395 – 1402, 2005.

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2:428 – 442, 2006.

[5] T.X. Han, H. Ning, and T.S. Huang. Efficient Nonparametric Belief Propagation with Application to Articulated Body Tracking. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 1:214– 221, 2006.

[6] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A Biologically Inspired System for Action Recognition. *Proc. IEEE Int. Conf. Computer Vision*, 2007.

[7] I. Laptev and T. Lindeberg. Space-time Interest Points. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 432 – 439, 2003.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004.

[9] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2118– 2125, 2006.

[10] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006.

[11] J.C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal Salient Points for Visual Recognition of Human Actions. *IEEE Trans. Systems, Man and Cybernetics Part B*, 36(3):710 – 719, 2005.

[13] R. Poppe. Vision-based human motion analysis: An overview. *Comp. Vision, and Image Understanding*, 108(1-2):4–18, 2007.

[14] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering Objects and their Location in Images. *Proc. IEEE Int. Conf. Computer Vision*, 1:370 – 377, 2005.

[16] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In *LNCS*, volume 4170, pages 127–144, 2006.

[17] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-Based Hand Tracking Using a Hierarchical Bayesian Filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(5):1372–1384, 2006.

[18] M.E. Tipping. The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, pages 652 – 658, 1999.

Determining Resource Needs of Autonomous Agents in Decoupled Plans

Jasper Oosterman [a] Remco Ravenhorst [a] Pim van Leeuwen [b]
Cees Witteveen [a]

[a] *Delft University of Technology, Algorithmics group, Delft*
[b] *National Aerospace Laboratory NLR, Amsterdam*

Abstract

At airports, the turnaround process of aircraft is crucial for a timely and efficient handling of air traffic. During this process, a number of services need to be provided during the time the aircraft is at the gate: e.g., de-boarding, cleaning, catering, fuelling, and boarding. These services are provided by different service providers (agents), who have to coordinate their activities in order to respect the turnaround time slot, the required service times and existing regulations. These temporal dependencies between services can be planned, but if disruptions occur re-planning is complex and often impossible. For this reason, in previous research a methodology and prototype have been devised to simplify the planning process by decoupling the overall plan into local plans for each agent. This paper builds on this research and introduces an important extension to the existing methodology and prototype: a new algorithm is introduced to take into account the minimal number of resources each service provider will need to accomplish its task.

1 Introduction

The central problem we discuss in this paper can be stated as follows. Given a specification C of temporal constraints for a set A of activities of a group of autonomous agents, *(i)* how to obtain for each agent i a set C_i of local constraints for its activities $A_i \subseteq A$, such that agent i can schedule its activities independently from the others, and *(ii)* how to provide each agent i with an estimation of the number of resources it will need to carry out its set of activities. Essentially, this problem can be viewed as a *coordination problem* since it requires the establishment of local temporal constraints for each agent in such a way that, while each agent can choose its own schedule independently from the others, the feasibility of the overall solution is still ensured.

In previous research, a methodology and prototype have been devised to solve this coordination problem applied to airport planning problems [8]. In a nutshell, in this research airport activities and their constraints were represented as a Simple Temporal Network (STN) and the temporal decoupling algorithm introduced by Hunsberger [5] was applied in order to break it up into several local (i.e. agent specific) temporal specifications. To illustrate the relevance of this approach, the *turnaround process* of aircraft was chosen as an application domain. The current paper builds on this previous work, and re-uses the application domain chosen in [8]. Of course, the results of the approach described here can be applied more generally.

In this paper, two important extensions are introduced to the existing decoupling methodology and turnaround prototype developed in [8]. First, a new algorithm is introduced to take into account the capacity - in terms of resources - of each ground handling agent for accomplishing its task. Second, before determining the order of tasks and consequently the capacity, we encompass the travel time between gates for each agent. This makes the application much more realistic, since the actual travel times between gates strongly influence the required capacity.

The turnaround process at airports concerns all activities that have to take place after an aircraft has arrived at its gate (the *on-block* time) until it departs from the gate (the *off-block* time). During this time passengers have to de-board and board, the aircraft has to be fuelled, catered and cleaned, technical inspection and maintenance have to be performed, amongst other things. These activities have to be performed by several autonomous agents (service providers) and are highly dependent on one another. For example, fuelling is only allowed after all passengers have deboarded and before new passengers have boarded.

Currently, for every type of aircraft, a turnaround plan is devised specifying for each service (catering, cleaning, boarding, fuelling, etc.) the exact time slot in which it should be performed. The collection of these plans constitutes the overall turnaround plan for an airport for a specific day. An example high-level entry of such a turnaround plan is: (KL1857, C06, 12:00, 13:28) where the flight number, the gate, and the on- and off-block time of a specific aircraft, respectively, is specified. At a lower level, all services that need to be planned between the on- and off-block times are listed, taking the temporal dependencies and minimum and maximum duration of each service into account.

To represent such a turnaround plan, often *Simple Temporal Networks* (STNs) are used [3]. An STN is a compact representation of all the (temporal) relationships between activities that need to be planned. In the aforementioned work [7, 8], a high-level STN was constructed of the overall domain. In a second step, a special Temporal Decoupling technique [5] was used to split the high-level STN S in a collection $\{S_i\}_{i=1}^{n}$ of sub STNs - one sub STN per service provider i. The outcome is that each of the STNs S_i contains a specification of the activities and temporal constraints for each service provider i. This service provider then is able to schedule its own activities without being forced to take into account the decisions of other agents: the temporal decoupling ensures that every combination of individually constructed schedules can always be merged into a global conflict-free schedule for the original global STN.

Although this technique offers the significant advantage of autonomous scheduling capability to the agents, the individual STNs do not provide information about the number of resources a service provider requires for carrying out its temporal plan in the STN. For example, from the global STN, a local STN can be decoupled for the fuelling service provider. Such a local STN contains a specification of all n fuelling activities the service provider has to perform on a certain day. In general the number m of fuelling cars the company has will be significantly less than n, and the fuelling company will be interested in a detailed specification of the *minimum* number of resources (e.g., vehicles) needed to perform all the activities. Moreover, each vehicle will need a specific amount of time to reach gate y after having fuelled an aircraft at gate x. Therefore, for each pair of gates x and y the minimum travel times $tt(x, y)$ need to be specified, leading to an additional set of temporal constraints specified in the local STN of the fuelling company.
Our resource consumption problem then can be stated as follows:

> *Given an STN containing the temporal constraints for n activities taking place at m different locations and each requiring one type of resource together with a specification of the travel times between the resource application locations, how can one determine the minimum number of resources needed to perform all activities satisfying the constraints?*

To answer this question, in Section 2 the necessary background on STNs and the Temporal Decoupling method is discussed. In Section 3, the algorithm to solve the resource consumption problem is detailed. Finally, in Section 4 some possible extensions of this solution are discussed.

2 Background

For each aircraft serviced at an airport a *turnaround plan* exists that specifies which activities (boarding, fuelling, cleaning, etc.) need to be performed when during the time the aircraft is at the gate/stand. In this section we will use a Simple Temporal Network (STN) to model a collection of turnaround plans and discuss the properties of such an STN in more detail. Next, we discuss a decoupling method that enables us to split a given STN in independent sub-STNs such that independently obtained solutions to these sub-STNs can be simply merged to obtain a solution for the original STN. The decoupling method will be used to split the global STN representing the total collection of turnaround plans into sub-STNs for each of the agents, allowing them to construct their operational plans independently of each other without violating any of the original global constraints.

2.1 Simple Temporal Networks

To model the turnaround plans we use a *Simple Temporal Network (STN)* [3]:

Definition 2.1 *A Simple Temporal Network S is a tuple $\langle X, C\rangle$ where X is a collection $\{x_0, ..., x_n\}$ of time variables, and C is a finite collection of constraints over these variables. Each constraint $c \in C$ is of the form $c_{ij} : x_j - x_i \leq b_{ij}$, for some $b_{ij} \in \mathbb{Z}$. The variable x_0, the temporal referential point, represents a special fixed time value, usually taking the value 0.*

To use STNs to specify temporal constraints on activities, every activity a_i is represented by two time variables (events) x_i and x_{i+1} indicating the starting and the finishing time of a_i, respectively.

In an STN S, for every pair of variables $x_i, x_j \in X$ there is a constraint $c_{ij} : x_i - x_j \leq b_{ij}$ and a constraint $c_{ji} : x_j - x_i \leq b_{ji}$. If these constraints are combined we obtain the *interval constraint* $-b_{ji} \leq x_i - x_j \leq b_{ij}$, also written as $I_{ij} = [-b_{ji}, b_{ij}]$. In this paper we will use both notations. As a special case we mention the interval constraint $0 \leq x_i - x_j \leq \infty$ specifying that x_j has to occur before x_i. Let us now present a simplified example of an STN in the turnaround domain.

Example 2.1 *Suppose we have a flight with an on and off-block time of 13:00 and 15:30, respectively. Therefore, all required ground services like fuelling, (de)boarding and cleaning have to be done between 13:00 and 15:30. It is required that deboarding has to start within 15 minutes after on-block and it takes at least 10 and at most 20 minutes. Fuelling can only start if deboarding has ended. Finally, we know that fuelling takes at least 20 and at most 40 minutes and has to be completed 30 minutes before off-block. We model these data in an STN by using the following nodes*

x_0 = *temporal reference point*	x_2 = *start deboarding*	x_4 = *start fuelling*	x_6 = *off-block time*
x_1 = *on block time*	x_3 = *end deboarding*	x_5 = *end fuelling*	

and the following constraints:

$60 \leq x_1 - x_0 \leq 60$	*on-block exactly 60 minutes after 12:00:00*
$150 \leq x_6 - x_1 \leq 150$	*the time between on- en off-block is 150 minutes*
$0 \leq x_2 - x_1 \leq 15$	*deboarding has to start within 15 minutes after arrival*
$10 \leq x_3 - x_2 \leq 20$	*duration of deboarding is between 10 and 20 minutes*
$20 \leq x_5 - x_4 \leq 40$	*fuelling takes 20 to 40 minutes*
$0 \leq x_4 - x_3 \leq \infty$	*fuelling starts after deboarding has ended*
$30 \leq x_6 - x_5 \leq \infty$	*fuelling has to completed at least 30 minutes before off-block*

Figure 1 contains a graphical representation (a Simple Temporal Network) derived from this STP.

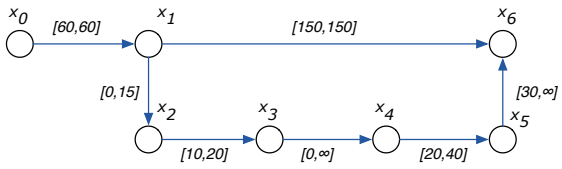


Figure 1: A Simple Temporal Network

The *solution* of an STN is a specification of suitable values for time variables $x_i \in X$:

Definition 2.2 *[5] A solution for an STN $S = \langle X, C\rangle$, where $X = \{x_1, x_2, \ldots, x_n\}$, is a complete assignment $\{x_0 = 0, x_1 = w_1, \ldots, x_n = w_n\}$, where each $w_i \in \mathbb{Z}$, such that all constraints $c_i \in C$ are satisfied.*

If such a solution exist, we say that S is *consistent*, else it is said to be *inconsistent*. There is an efficient algorithm to check (in)consistency of an STN based on the direct labelled graph representation $G_S = \langle N_X, E_C, l\rangle$ of $S = \langle X, C\rangle$, where N_X is a set of nodes n_i representing the time points x_i, E_C is a set of directed arcs, where $e = (n_i, n_j) \in E$ has label $l(e) = b_{ji}$ whenever $c_{ji} : x_j - x_i \leq b_{ji}$ occurs in C. If the labels $l(e)$ are interpreted as distances, the well-known $O(n^3)$ Floyd-Warshall [2] All-Pair-Shortest-Path

(APSP) algorithm can be used to determine the distance $d(i,j)$ between all nodes n_i and $n_j \in N_X$. Now, the inconsistency of S can be easily decided by checking whether $d(i,i) \geq 0$ holds for all nodes $n_i \in N_X$.

The graph G_S^D obtained by applying the APSP algorithm to the STN G_S containing all shortest distances between the nodes in N_X and therefore specifying the *tightest* constraints between the time variables in X, is called the *d-graph* associated with S. Note that the d-graph is a complete graph and contains, for every node pair, one edge with the distance between them as label.

2.2 The Temporal Decoupling Method

We model the collection of turnaround plans using as an STN. This STN contains the specifications of all activities to be performed by the different agents (service providers) whilst on-block. In order to provide a solution for the complete turnaround process we need to specify the values of all begin and endpoints of these activities. To ensure a valid solution this seems to require either a centralized solution process or a rather elaborate coordination process requiring negotiation between the different agents involved. There is, however, another possibility. This requires a modification of the original STN S such that

- The STN S is split into k sub STNs $S_1, \ldots, S_k$ according to the activities belonging to the k participating agents (service providers);
- Each of the agents is allowed to solve its own sub STN S_i and to specify its own solution s_i for it;
- whatever solutions s_i are chosen by the agents, their merge $s = (s_1, s_2, \ldots, s_k)$ always constitutes a valid solution to the overall STN S.

This so-called *temporal decoupling method* specified by Hunsberger [5] can be defined in a more precise way as follows:

Definition 2.3 *(z-partition)*
Let X, X_X and X_Y be collections of timepoint variables. We say that X_X and X_Y, z-partition X if:

- $X_X \cap X_Y = \{z\}$, *where* $z = x_0 = 0$
- $X_X \cup X_Y = X$

Definition 2.4 *(Temporal Decoupling)*
A temporal decoupling of the STN $S = \langle X, C\rangle$ is a pair of STNs $S_X = \langle X_X, C_X\rangle$ and $S_X = \langle X_Y, C_Y\rangle$ such that:

- *S_X and S_Y are consistent,*
- *X_X en X_Y z-partition X, and*
- *the merging of solutions for S_X and S_Y always is a solution for S.*

The definition of the z-partition and the temporal decoupling for more than two sets is analogous to that for 2 sets. In our application we choose the z-partition in such a way that all time points belonging to activities of one service provider occur in the same z-partition. We will now give an example of z-partitions and removing the dependencies.

Example 2.2 *Assume we have a fuelling company and a boarding company who have agreed to perform two tasks. The first task has start and end time x_1 and x_2, and the second task x_3 and x_4, respectively. The fuelling company performs the first task and the boarding company performs the second one. Both tasks are allowed to start between 15:00 and 16:45, but the fueling company should have finished when the boarding company starts. Both tasks take exactly 15 minutes to complete. The joint STN is shown in the first part of Figure 2.2. Here, we take x_0 as the time reference point (X_0=15:00).*

Both companies don't want to coordinate their activities.Therefore we will choose the z-partition in such a way that the fuelling and boarding company are in different blocks. Note that both z-partitions only have x_0 in common (See Figure 2.2). To remove the dependency between the companies we have to satisfy the constraint $0 \leq x_3 - x_2 \leq \infty$ between them. This is accomplished by changing the constraints $0 \leq x_1 - x_0 \leq 105$ and $0 \leq x_2 - x_0 \leq 105$ as shown in the second part of Figure 2.2. The effect of this change is that the inter-company constraint $0 \leq x_3 - x_2 \leq \infty$ is always satisfied and can be removed. As a consequence, the time interval in which both companies can start their task is smaller, but they can perform their tasks independently from one another.

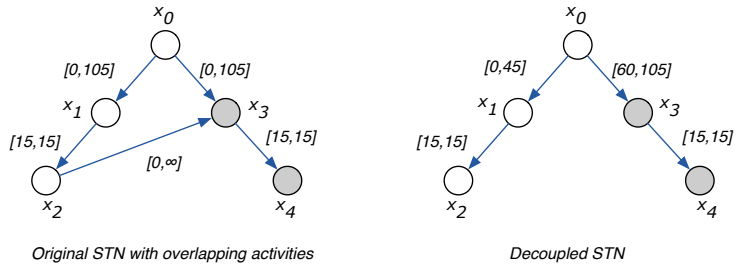


Figure 2: Decoupling of an STN

In [4, 5] an algorithm is described that can solve Temporal Decoupling problems[1].

3 Determining the number of resources needed

With the help of the temporal decoupling algorithm an independently schedulable temporal plan per agent (service provider) can be achieved. In general such a schedule will also include the number of resources needed in order to complete all the activities specified in the temporal plan of the service provider. For example, agents such as the fuelling or catering company need vehicles as resources to service an aircraft. It is possible for one resource to service multiple aircraft if the activities take place after each other and there is sufficient time between them to travel. Hence, we need to take into account the (travelling) time between the activities to be performed by each agent. To give an example, let us consider the temporal plan of the fuelling agent. In such a plan the Earliest Starting Time ($EST(a)$), Latest Starting Time ($LST(a)$), Earliest End Time ($EET(a)$) and the Latest End Time ($LET(a)$) are specified per activity a. Table 1 shows an example in which for each flight it is indicated at what time the fuelling agent must start (earliest) and when the agent must end (latest).

In order to determine the number of fuelling vehicles for this agent A_{fuel}, we model its set of activities A specified in the decoupled temporal plans as the set of nodes V in a graph $G_A = \langle V, E \rangle$. There is a directed edge between two nodes $a_i, a_j \in V$ if it is possible to accomplish activity a_j after a_i with the same resource. Here, we can use two methods to define this relation: the *minimum* and the *maximum* method.

- minimum method: $(a_i, a_j) \in E$ iff $EET(a_i) + distance(a_i, a_j) \leq LST(a_j)$
- maximum method: $(a_i, a_j) \in E$ iff $LET(a_i) + distance(a_i, a_j) \leq EST(a_j)$

Here, $distance(a_i, a_j)$ is a table with travel times (distances) between the gate for a_i and the gate where a_j has to be performed. In the next example we show how to construct such a graph using the maximum method.

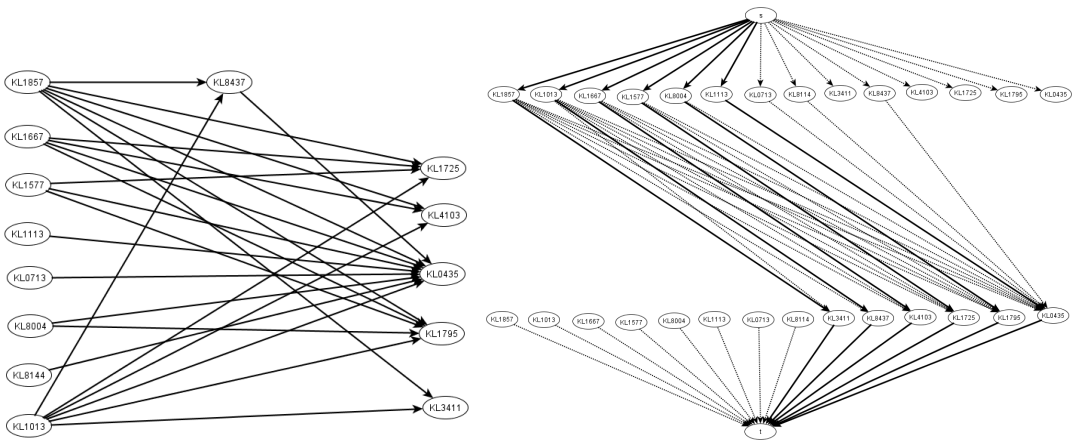
Example 3.1 *We construct the graph $G_A = \langle V, E \rangle$ for the fuelling agent. First, we need its decoupled plan with a list of activities with all the flights and the EST, LST, EET and LET for the fuelling service for these flights. Table 1 shows a part of this list. We also need a table with the travel times (distances) between gates. We check if for every possible pair of activities a_i and a_j the equation $LET(a_i) + distance(a_i, a_j) \leq EST(a_j)$ holds. If so, we add a directed edge (a_i, a_j) to G_A. If not, no edge is added. For example, if we check for flight KL1857 and KL1577, we have 13:28:31 + 00:02:00 > 12:25:23. It follows that it is not possible to service these two flights after each other and consequently no edge is added. Applying this method to all pairs results in Figure 3(a).*

In Figure 3(a) there is a path from flight KL1857 via KL8437 to KL0435. This means that only one resource is needed to service these three flights. In order to determine the minimum number of resources needed, we now have to find out how many paths we need if we want to cover all nodes exactly once. In the literature, this problem is called the *Minimal Node Disjoint Path Cover* problem (e.g., [6]). In general, this problem is intractable, because the decision variant (*Node Disjoint Path Cover*) is a special case of the NP-complete

[1] This algorithm uses a so-called *greediness* factor which is used to select the balance between flexibility of the solution and computation time

Table 1: Decoupled plans for fuelling service

Call sign	Gate	EST	LST	EET	LET
KL1857	C06	12:00:00	12:00:00	12:12:00	13:28:31
KL1013	B13	12:10:00	12:10:00	12:22:00	13:39:37
KL1667	D87	12:14:25	12:14:25	12:45:25	13:59:31
KL1577	F04	12:25:23	12:25:23	12:56:23	14:10:37
.....	.....	.....	.....	.....	.....



(a) Graph G_A for the fuelling company created with the maximum method.

(b) Flow graph G_F constructed from the graph G_A in Figure 3(a). Bold lines indicate flow.

Figure 3: Working examples

HAMPAD problem (Covering all nodes with one path). In our case, however, the graph G_A is acyclic[2], which, as we will show, implies that the problem can be solved in polynomial time.

To solve this problem, we will show here that the minimum number of paths can be determined by reducing the problem to a polynomial solvable *Maximum Flow* problem[2]. First, we will construct a flow graph G_F from the graph G_A presented in Figure 3(a). Second, we will prove that a solution of this *Maximum Flow* problem can be easily converted into a solution of our Minimal Node Disjoint Path Cover problem, i.e., the minimum needed capacity problem.

First, we construct from the directed graph $G_A = (V, E)$, created with maximal or minimal method, a new (flow) graph $G' = (V', E')$ as follows. Let $V = \{x_1, x_2, ..., x_n\}$.

- $V' = \{s, x_1, x_2, \ldots, x_n\} \cup \{y_1, y_2, \ldots, y_n, t\}$, where s (the source) and t (the sink) are two nodes not occurring in V and for every $x_i \in V$, y_i is a new node in V';
- $E' = \{(s, x_i) : x_i \in V\} \cup \{(y_i, t) : x_i \in V\} \cup \{(x_i, y_j) : (x_i, x_j) \in E\}$.

Performing this transformation on the graph G_A, we get the flow graph G_F depicted in Figure 3(b). The top row is called the X-row with the x_i nodes and the bottom row of nodes is the Y-row of copies y_i. We give each edge $e \in E'$ a capacity of 1. Executing a Maximal Flow algorithm on G_F gives the maximum amount of flow in the graph and the edges through which it flows (See Figure 3(b), the edges in the maximal flow are marked bold). The flow f is a maximum flow with value 6.

We now show that, in general, the value of the maximal flow in G_F determines the solution specifying the minimum amount of resources needed (i.e. the solution to the minimum disjoint path cover problem)

[2]Note that the graph is always transitive: if there exists an edge between node a and b then, according to the maximal method, $a_{let} + distance(a, b) \leq b_{est}$ holds. If there also exists an edge (b,c) then $b_{let} + distance(b, c) \leq c_{est}$ also holds. By definition $b_{est} \leq b_{let}$, from which follows that $b_{est} + distance(b, c) \leq c_{est}$ and thus $a_{let} + distance(a, b) + distance(b, c) \leq c_{est}$ and $a_{let} + distance(a, b) \leq c_{est}$. This transitivity of the graph ensures that the requirement that each flight is serviced (fuelled) exactly once doesn't restrict the solution set.

Proposition 3.1 *Given a graph $G = (V, E)$ and the constructed flow graph $G' = (V', E')$, let f be the value of the maximal flow of G'. Then the minimum number of paths needed for a disjoint path cover of G is $|V| - f$.*

Proof See Appendix.

Applying the Maximum Flow algorithm to our example in Figure 3(b), we find a maximum flow of 6. Thus, the minimum number of resources needed to carry out all activities is 8 (the number of activities (14) − the maximum flow).

Note that in addition to this solution (a path cover) our algorithm also determines *which* flights can be serviced by the same resource. Each edge with positive flow is a part of a path. In Figure 3(b) these edges are marked bold. Constructing the paths is now easy. We take for each edge (x_i, y_j) with positive flow the edge (v_i, v_j) in G and add it to the path cover. Succeeding nodes are on the same path. Nodes not belonging to a path belong to their own path with length 0. In Figure 3(a) the paths are the following ones:

- Paths with length 1: $\{(KL1857, KL3411)\}$, $\{(KL1013, KL8437)\}$, $\{(KL1667, KL4103\}$, $\{(KL1577, KL1725)\}$, $\{(KL8004, KL1795)\}$ and $\{(KL1113, KL0435)\}$;
- Paths with length 0: KL0713 and KL8114.

In Figure 4 the resources needed are plotted against time. This figure demonstrates that the maximum capacity of 8 vehicles is only needed between 13:00 and 16:00 - not for the complete time period.

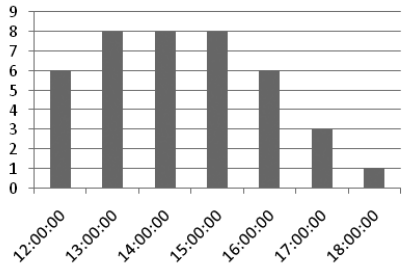


Figure 4: Number of resources needed dependent upon time

4 Discussion

In this paper we discussed a multi-agent approach to coordination with an application to the turnaround process on airports. In particular, we discussed a temporal decoupling method that could be used to offer service providers at an airport a stand alone specification of their activities which enables them to plan/schedule their activities independently of the planning activities of other agents. Given this decoupling of turnaround plans, we presented an algorithm to determine the number of resources needed to execute all the activities in the decoupled plan, based on the (earliest en latest) start and end times and the travelling time between activities. The time complexity of this algorithm is rather modest: it is bounded above by the complexity of the Maximum Flow algorithm ($O(|V||E| \log(|V|^2/|E|))$ and lower bounded by $O(|V|^2)$. [3]

An advantage of this algorithm is that it not only gives the amount of needed resources, but also where and when these should operate. For example, figure 4 shows that the minimum capacity of 8 is only needed for tree hours on the whole day and that the rest of the time the aircraft can be serviced with less resources. It is possible to have different path covers with the same maximum value. For agents these path covers can be more or less efficient, because the total distance driven by resources could be lower in another path cover. It would be an improvement to use a weighted Maximum Flow algorithm, where the weight on an edge is the distance between the two activities. Further research can also be done on optimizing the selection and length of paths in the path cover in order to divide the load between individual resources.

[3] In [1] the fastest algorithms available up till now are presented.

In this paper we made a few assumptions in order to reach an efficient solution. Our methods for determining the edges use the duration of an activity as one value (20 or 30 minutes) instead of an interval (20 to 30 minutes). We did provide maximal and minimal methods which give the lower and upper bound of the resources needed. We also assumed that an aircraft needs the same kind of service no matter at what gate it is serviced. For example, each aircraft in our algorithm needs to be fuelled with a fuelling vehicle, while some airports have fuel depots on certain gates and no vehicle is necessary. Another assumption is that resources are always able to service. For example, in our algorithm a fuelling vehicle is able to service an unlimited amount of aircraft without refuelling itself. An important extension to the algorithm would be the adding of these specific constraints which give a much more realistic view on resource movement.

References

[1] B. V. Cherkassy and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. In *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 157–171, London, UK, 1995. Springer-Verlag.

[2] Th. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[3] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95, 1991.

[4] L. Hunsberger. Algorithms for a temporal decoupling problem in multi-agent planning. In *Eighteenth national conference on Artificial intelligence*, pages 468–475, Menlo Park, CA, USA, 2002. AAAI.

[5] L. Hunsberger. *Group Decision Making and Temporal Reasoning*. PhD thesis, Cambridge, MA, USA, 2002.

[6] J. M. Kleinberg. *Approximation algorithms for disjoint paths problems*. PhD thesis, 1996.

[7] P. van Leeuwen. The coordinated airport through extreme decoupling. Technical report, Nationaal Lucht en Ruimtevaart Laboratorium (NLR), 2007.

[8] P. van Leeuwen, L. I. Oei, P. Buzing, and C. Witteveen. Adaptive temporal planning at airports. In *Proceedings of the International Multi-conference on Computer Science and Information Technology*, Wisla, Poland, 15 - 17 October, 2007.

A Proof of Proposition

Proposition A.1 *Given a graph $G = (V, E)$ and the constructed flow graph $G' = (V', E')$, let f be the value of the maximal flow of G'. Then the minimum number of paths needed for a disjoint path cover of G is $|V| - f$.*

Proof We show that G' has a flow with value $|V| - k$ if and only if k is set of disjoint paths covering V. From this correspondence it follows immediately that maximal flows correspond to minimum disjoint path covers.
($\Leftarrow$) Given graph G with a path cover consisting of k disjoint paths, there are k nodes as the starting point of a covering path and $|V| - k$ which are not. Because the paths are node disjoint, each of these $|V| - k$ nodes has a unique predecessor. We construct a flow graph G' according to the method described earlier. For each node v_{i+1} with predecessor v_i there is an edge in G' between x_i from the X-row and y_{i+1} from the Y-row over which one unit of flow can be pushed. Because each node has a unique predecessor, the capacity constraint holds because each node x_i has at most one outgoing flow and each y_{i+1} has at most one incoming flow. So G' has as flow of $|V| - k$.
($\Rightarrow$) Consider flow of size $|V| - k$ in the graph G' constructed from graph $G = (V, E)$. It is not difficult to see that this implies that there are k disjoint paths from the source s to the sink t. Hence, there are k disjoint nodes (capacity is 1) in the X-row from which 1 unit of flow flows to a unique node in the Y-row. From this follows that there are k nodes in G that are a successor of some node on a flow path, and $|V| - k$ that are not. Clearly, these $|V| - k$ nodes are the starting point of a covering path. Therefore, there are $|V| - k$ paths in the path cover for G.

Categorizing Children
Automated Text Classification of CHILDES files

Rob Opsomer [a] Petr Knoth [b] Freek van Polen [c] Jantine Trapman [c]

Marco Wiering [d]

[a] *Ghent University, Faculty of Engineering*
[b] *The Open University, Knowledge Media Institute*
[c] *Utrecht University, Department of Philosophy*
[d] *University of Groningen, Department of Artificial Intelligence*

Abstract

In this paper we present the application of machine learning text classification methods to two tasks: categorization of children's speech in the CHILDES Database according to gender and age. Both tasks are binary. For age, we distinguish two age groups between the age of 1.9 and 3.0 years old. The boundary between the groups lies at the age of 2.4 which is both the mean and the median of the age in our data set. We show that the machine learning approach, based on a bag of words, can achieve much better results than features such as average utterance length or Type-Token Ratio, which are methods traditionally used by linguists. We have achieved 80.5% and 70.5% classification accuracy for the age and gender task respectively.

1 Introduction

In this paper, state-of-the-art text classification methods are applied to two tasks: categorization of transcribed children's speech according to gender and age. Various machine learning techniques from the field of text classification were applied. We summarize some widely used machine learning methods such as k-Nearest Neighbours, Neural Networks, Support Vector Machines and Boosting. We compare their performance on the two classification tasks. The text classification methods are based on the bag-of-words approach. In this approach, one looks at the frequency of words in a text, without considering their order.

The text classification methods are compared for their accuracy with traditional measures used in linguistics, such as the average utterance length and the Type-Token Ratio. Despite the fact that these measures are considered as standard measures, they are, as indicators of (morpho-)syntactic complexity, widely discussed for their reliability.

2 Data Set

Both the traditional measures and the machine learning methods are applied on a data set from the CHILDES (CHIld Language Data Exchange System) [1] Database [8]. This data set, named *Manchester*, contains speech of 12 different British children.

Every child was recorded approximately the same number of times, on average about 65 times. That way, we have a data set of 806 conversations. The distribution of the conversations over age and gender is quite equal. For the age classification task we have drawn the boundary at age of 2.4. This decimal number is both the mean and the median of the age in our data set.

Figure 1 shows the distribution of age and gender in our data set. The data set contains slightly more recordings from girls than boys.

[1] See `http://childes.psy.cmu.edu/`

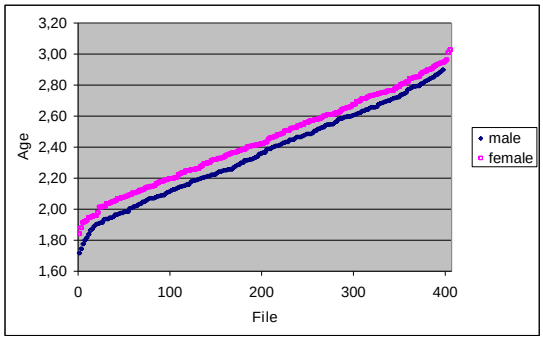


Figure 1: Manchester: female and male age distribution, in decimal years

2.1 Original document representation

In its original form, the Manchester data set is a highly structured data set, with all conversations transcribed in CHAT format for various linguistic layers. In general, each file contains two parts: a header and a body. The header contains metadata about the child, the other interlocutors, date of recording etc. The body consists of the transcription of the actual conversation. Apart from the utterances of the child, utterances of other interlocutors are included. Each utterance is displayed on a separate line and provided with an extra, morphological layer. In the example below the '*'-marked lines are utterances from mother and child respectively. The %mor-layers contain POS-tags of the words. The example shows also a typical transcription code: the part between brackets indicates that the word is guessed by the transcriber but omitted in the actual utterance.

*MOT: again ?
%mor: adv|again ?
CHI: where [0are] ?
%mor: adv:|where n|car-PL ?

There are a lot of such special transcription codes in the transcriptions. Examples are the use of 'xxx' and 'www' for unintelligible speech, and the insertion of the '+' character in compounds such as *ice+cream*. Even gestures and noises are included. All these codes are exhaustively described in the guidelines for the CHAT transcription: *Codes for the Human Analysis of Transcripts*.

2.2 Preprocessing

From the header the data about age and gender were extracted. All other metadata was discarded.

In the body, utterances from all other participants were discarded. Moreover, the morphological layers were also removed. Initially, we maintained two data sets, one with *plain* and one with morphological data. Since initial experiments showed inferior results with the morphological data, we have only continued research using the plain data set.

All special transcription codes were also discarded, and compounds were rewritten as single words (e.g. *icecream*). After preprocessing each file contained on average 616 words.

Original sentence:	**Processed sentence:**
CHI: <me go> [/] me go xxx .	me go

One of the stages within language acquisition is the differential phase which begins at the age of $2\frac{1}{2}$ years[2] [5]. In this phase a child starts to use inflected verbs (instead of infinitives), auxiliaries, modals and

[2]For some children this phase might start a little later

function words among others. The children involved in our project are at an age where this differential phase might start. This means that the use of suffixes and function words should improve the performance of the age classification. Hence, unlike regular text classification, stemming and removal of function words is *not* applied to our data.

3 Document Representation

After the initial preprocessing phase, the documents are converted into a form that can be used by the clasiffication algorithms. The documents are represented either by *naive features* or a bag-of-words.

3.1 Naive Features

The first conversion method is based on the extraction of a number of naive features from each document, such as the average sentence length or the amount of different words in the document. The latter can be normalised by the document length. In that case it is called the Type-Token Ratio. These measures are traditional methods used by linguists to acquire an indication of a child's language development.

3.2 Bag of words

The second way to represent documents is having each document represented by a feature vector of size n, where n is the number of entries in the dictionary. The values of the feature vector can be either 1) word frequencies denoting how often each word occurs in the document, 2) word frequencies as a percentage of the total number of words in the document, 3) binary values denoting for each word whether or not it occurs in the document, or 4) the more sophisticated term frequency-inverted document frequency (tf-idf) measure [9].

The dictionary consists of a number of words. The basic way to construct the dictionary, is to simply use all words that occur at least once in the training set. Since this generally yields a dictionary that is very large, a simple dimensionality reduction technique is applied. In this technique, only the words that occur at least τ times in the training set are included in the dictionary. Here τ is a threshold that can be varied at will. Experimental evidence suggests that using only the top 10 percent of the most frequent words does not reduce the performance of classifiers [3]. In our experiments, we have experimeted with the value of τ between 0 and 40.

4 Classification Algorithms

For the classification of our documents we use the following state-of-the-art methods for text classification [2]: k-Nearest Neighbours (k-NN), Support Vector Machines (SVMs), neural networks (NNs), and the boosting algorithm.

4.1 k-Nearest Neighbours

The k-NN method is easy and straightforward to implement, and it has been reported before that it is quite effective in text classification [9].

For classification of a certain document, the k documents that are most similar to this test document are selected from the training set. The categorization status value (csv) of the test document is then calculated as:

$$csv = \frac{1}{k} \sum_{i \in neighbours} csv_i \tag{1}$$

where the csv of a training document is always -1 or 1. The csv numbers represent a class: for example -1 is female, 1 represents male. A slightly modified variant for age classification:

$$csv = 1 - \frac{1}{k \times ageMean} \sum_{i \in neighbours} age_i \tag{2}$$

where the csv of a training document is always -1 or 1. For age classification the csv is a weighted function shown in equation (2). Since our classification tasks are binary, the csv is then rounded to either -1 or 1.

In order to determine which training documents are most similar to a test document, one needs a distance function. In this work, the Euclidean distance and Cosine similarity are used.

4.1.1 Cosine Similarity

The Cosine similarity is a common vector based measurement calculating the similarity between two vectors on a $[0, 1]$ scale. Similarity of 1 means for two vectors to be either identical or different by a constant factor. Given feature vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ the Cosine Similarity is defined as:

$$cos(X, Y) = \frac{x_1 \cdot x_2 + \cdots + x_n \cdot y_n}{\sqrt{x_1^2 + \cdots + x_n^2} \cdot \sqrt{y_1^2 + \cdots + y_n^2}} \tag{3}$$

4.2 Support Vector Machines

SVMs are very powerful classifiers for text classification [9]. The idea of SVM is based on simultaneous minimization of classification error and maximization of the geometric margin [10]. The original optimal hyperplane algorithm was proposed by Vladimir Vapnik in 1963 and involved a linear classifier. This was extended in 1992 applying the *kernel trick* to maximum-margin hyperplanes. Since that extension, the transformation may be non-linear in the original input space. Some common kernels are Radial Basis Function (RBF), Polynomial and Sigmoid.

In this project we use the *LIBSVM* package. LIBSVM is an integrated software package for support vector classification, regression and distribution estimation [1]. In this package, manually tuning of parameters is not necessary. By default, grid-search, which tries different combinations of parameters C and γ using cross-validation, is used to detect parameters that maximize accuracy. Note that only the RBF kernel is used in the experiments, because the linear kernel is a special case of RBF and the polynomial kernel has more numerical difficulties, since kernel values may go to infinity [6].

4.3 Neural Networks

In our work, we have used both linear (one-layer) and non-linear (multi-layer) neural networks.

In the linear NN, often called perceptron, the input layer consists of n input units, where n is equal to the amount of features each document is represented by. Each input unit is connected to one single output unit. The activation of the output unit is the weighted sum of the activations of the input units. Since the two classification tasks at hand are both binary, just the sign of the output unit suffices to determine which class it picked. The delta learning rule is applied to train the network. Learning is only performed when the network outputs the wrong sign, and if it does, it tunes the output toward values of 1 and -1.

In the multi-layer NN, there is also an input layer of size n, where n is equal to the amount of features each document is represented by. However, this time each input unit is connected to ten hidden units, whose activations are determined by a *sigmoid* function over the weighted sum of the input activations. These ten hidden units are connected to a single output unit. The activation of the output unit is the weighted sum of the activations of the hidden units. Backpropagation is performed to train the network. As with the linear network, learning (backpropagation) is only performed when the sign is incorrect.

4.4 Boosting

In the boosting method one iteratively trains multiple weak classifiers, each focusing on the prevention of mistakes the previously trained classifiers made. The actual classification task is done by combining the votes of the different classifiers and weighing them by the error they had on the training set. The boosting algorithm we used is called AdaBoost [4]. We have used the boosting algorithm in combination with linear neural networks.

	Gender	Age
k-NN	59.2%	76.2%
SVM	70.5%	80.5%
Linear NN	70.0%	74.6%
Naive features	50.0%	69.2%

Table 1: Accuracy results for the age and gender tasks

5 Experimental Results

All classifiers were extensively tested on a wide range of parameters. We have tried to determine the optimal parameters for all classifiers. The results can be found in table 1.

All results are presented as accuracy scores. Cross-validation is used in our experimental set-up. More specifically, a *leave-one-child-out* approach is applied. This means that one child is held out from the data set, and the classifier is trained on the rest of the documents. The resulting classifier is then tested on the held out child. This process is repeated for each child in the data set. The results are averaged over all children.

For the bag of words approach, results are discussed for the different classifiers. The naive feature results were obtained with the k-NN algorithm. All classifiers achieved very similar results with the naive features. If not mentioned, results describe bag of words experiments.

5.1 Gender Classification

The best results were obtained with SVMs, giving a score of 70.5%. A wide range of $\tau \in \{0, \ldots, 40\}$ produced similar results. This shows that for gender the most frequent used words are the most informative.

Linear NNs also performed quite good on this task. An optimal result of 70.0% was obtained using term frequencies, a learning rate of 0.0001, $\tau = 30$ and 200,000 randomly sampled training examples. However, note that this result is an average of eight trials. We noted some random fluctuations in the results of the networks. This is probably due to the fact that the training examples are sampled randomly, and that the weights of the network are initialized randomly. Also, the dimensionality of the task is quite high, while the amount of training data is rather small. However, it is unclear why this influences the results so much.

The non-linear NNs and the boosting algorithm were even more unstable, probably because of the more complex nature of these algorithms. Therefore, we don't include any results of these algorithms, although there were some promising results.

k-NN produced quite disappointing results for this task. A maximum score of 59.2% was obtained, with $k = 7$, no dimensionality reduction, tf-idf features and the cosine similarity function.

As expected, the naive features produced inferior results for the gender classification task. With accuracies around 50%, it didn't perform better than the random classifier.

5.2 Age Classification

Again, the best results were obtained using SVMs. The optimal result here is 80.5%, obtained with tf-idf features. A wide range of $\tau \in \{0, \ldots, 40\}$ produced similar results. This shows that for age the most frequent used words are the most informative.

With linear NNs, we obtained a best result of 74.6% with term frequencies, $\tau = 50$, a learning rate of 0.00005 and 500,000 randomly sampled training examples. Note that we averaged those results over 8 trials. Again, the non-linear NNs and the boosting algorithm produced very unstable results.

k-NN performed quite well on this task. An optimal result of 76.2% was obtained with $k = 3$, the modified csv computation (see section 4.1), $\tau = 2$, binary features and an Euclidean distance function. This result is very robust to changes in k and τ. The combination of tf-idf and cosine similarity produced also similar results.

With the naive features, an optimal result of 69.2% was obtained. Average sentence length (and not the Type-Token Ratio) seemed to be the most informative about the children age.

6 Conclusions and Discussion

Very good results were obtained with support vector machines. For the age task, those results are very robust across children. All children achieved accuracy higher than 60%. However, for the gender task, we have seen that boys are in general classified worse than girls. Some boys are even most of the time classified as girls. Similar observations hold for all classifiers. It would be very interesting to find out why this is the case.

When we look at the accuracies we have achieved (70.5% for gender, 80.5% for age), we can theorize what kind of implications they have. For instance, a 70.5% accuracy on gender can be interpreted as indicating that it is possible to determine whether speech comes from a boy or a girl by just looking at how often certain words are used. This observation supports the claim that boys and girls have different vocabularies already at such a young age. Also for age, an 80.5% accuracy indicates that it is very well possible to determine the age of a child only by looking at its speech (under the condition that the child is in the age stage where language acquisition plays a role). Though this is less remarkable than being able to determine gender, maybe it can be used to determine the rate at which a child is developing its cognitive skills. If a child is classified as being younger than it actually is, this may mean that the child is behind in language development.

When talking about current results, we should take into account that some conversations are very short (especially with the younger children), and are thus almost impossible to classify well. A possible solution would be to concatenate some of the conversation files. We should also take into account that some children are maybe very bad representatives of their class. For example when classifying children near the age pivot, this task can become almost impossible. For this reason, it would be interesting to make an experiment that would measure the accuracy of humans on our tasks. That would bring some real world comparison to the presented machine learning techniques.

A number of interesting things can be noted about the neural networks and the boosting algorithm. First off, most of the time each document is represented by a very large amount of features (1,500 or more). Because there are about 650 training documents every time, the neural networks are very likely to suffer from overfitting. Especially the multi-layer neural networks are very prone to overfit on the training data. This could be corrected by choosing a lot smaller feature set, in a more sophisticated way than just removing all words that do not occur more than a certain number of times.

What applies for the neural networks algorithm, also applies for the boosting algorithm, since we use linear NNs in the boosting algorithm. If the initial classifiers are trained too long, they will overfit and achieve (near-)100% accuracy on the training set. The succeeding classifiers will then focus on just very few examples, heavily overtraining on them. As a result of this, one will observe that the amount of samples from the training set for the boosting algorithm is much lower than the amount of samples for the networks. We observe that for the optimal setting we have found, changing the amount of classifiers reduces the accuracy. This indicates that the choice of the number of classifiers is very important especially when applying the boosting algorithm.

7 Future Research

For future research, there are quite a few daunting and promising options. The most promising one for the improvement of the results is probably to work on the features. For example, a selected set of function words or n-grams of parts of speech could be used. These features were already applied in [7], revealing distinctive differences between male and female (adult) authors. Some of our preliminary experiments have showed that using part-of-speech tags for the age task does produce good results. However, these results were never significantly better than the results obtained using the plain utterances.

A combination of plain and morphological data could also be interesting. One could also try a combination of children and parental speech. In order to do that, it should be established whether parental speech really gives a significant indication of the child's age or gender. There are already assesments from some of our preliminary experiments that it does.

Furthermore, it would be interesting to tackle the scarce data problem we have to deal with. More sophisticated dimensionality reduction techniques could be used, such as odds-ratio [9] or Latent Semantic Indexing. Another thing to consider is to search for more data. We tried the latter, but it proved to be fairly difficult to find another good data set.

Another thing that would probably be an easy way to improve our results for the gender classification task would be to drop the leave-one-child-out approach, and to adopt a leave-two-children-out approach. With the leave-one-child-out approach, the training data always consists of 6 children of the opposite gender of the child we want to classify and 5 children of the same gender, implying a bias for the opposite gender. For the age task, it might be interesting to cast the task as a regression task, instead of a classification task.

Finally, what might be interesting, is to discuss the sociological implications (discussed in the previous section) with linguists or specialists in the field of language acquisition of children. Perhaps, they have interesting ideas to improve the results. Or maybe, they can provide at least a theoretical foundation for some of the results we have obtained. Moreover, they could use the results of these experiments in their own theoretical and practical work. For this purpose, it would be interesting to inspect the words (or parts of speech) that distinguish between the classes.

References

[1] C.C. Chang and C.J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001.

[2] M. Cord and P. Cunningham, editors. *Machine Learning Techniques for Multimedia: Case Studies on Organization and Multimedia*, volume XVI of *Cognitive Technologies*. Springer Verlag, Berlin, 2008.

[3] R. Feldman and J. Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, December 2006.

[4] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999.

[5] S. Gillis and A. Schaerlaekens. *Kindertaalverwerving: Een Handboek voor het Nederlands*. Martinus Nijhoff, Groningen, 2000.

[6] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2004.

[7] M. Koppel, S. Argamon, and A. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(3), 2003.

[8] B. MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition, 2000.

[9] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[10] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

A Neural Network Based Dutch Part of Speech Tagger

Mannes Poel Egwin Boschman Rieks op den Akker

University of Twente, Dept. Computer Science,
P.O. Box 217, 7500 AE Enschede, The Netherlands
{mpoel,infrieks}@cs.utwente.nl

Abstract

In this paper a Neural Network is designed for Part-of-Speech Tagging of Dutch text. Our approach uses the Corpus Gesproken Nederlands (CGN) consisting of almost 9 million transcribed words of spoken Dutch, divided into 15 different categories. The outcome of the design is a Neural Network with an input window of size 8 (4 words back and 3 words ahead) and a hidden layer of 370 neurons. The words ahead are coded based on the relative frequency of the tags in the training set for the word. Special attention is paid to unknown words (words not in the training set) for which such a relative frequency cannot be determined. Based on a 10-fold cross validation an approximation of the relative frequency of tags for unknown words is determined. The performance of the Neural Network is 97.35%, 97.88% on known words and 41.67% on unknown words. This is comparable to state of the art performances found in the literature. The special coding of unknown words resulted of an increase of almost 13% for the tagging of unknown words.

1 Introduction

A Part-of-Speech (PoS) Tagger is a system that assigns the correct Part-of-Speech or word class to each of the words in a document. Classical parts of speech are noun and verb, and a few others, but nowadays Part of Speech tag sets sub-divide these general word classes into smaller ones, such as proper names, singular nouns, mass nouns and plural nouns. Part of Speech tag sets typically contain from a little over twenty to more than a few hundred of different word classes. PoS tagging is a non-trivial task because most words are ambiguous: they can belong to more than one class, the actual class depends on the context of use. In this paper we approach the PoS tagging task using Neural Networks. For training and testing we use the Corpus Gesproken Nederlands (CGN - Spoken Dutch Corpus) a large morpho-syntactically annotated corpus. Figure 1 shows that the more common a word occurs in this corpus the more likely it is ambiguous. More details of this corpus are given in section 1.1. The data points are an average of the percentage of ambiguous words around that frequency. PoS tagging, or word class disambiguation, is the process of finding out the right word class for these ambiguous words. The result is then added as a label or 'tag' to the word. PoS tagging is often only one step in a text processing application. The tagged text can be used for deeper analysis, for example for chunk parsing or full parsing. Because the accuracy of the PoS tagging greatly influences the performance of the steps further in the pipeline [4], the accuracy of the PoS tagger is very important. In general PoS tagging can be seen as a sequential supervised learning problem [6] and various models for supervised machine learning have been applied to the problem of PoS tagging; memory based learning [5] , transformation rule based learning [3], (Hidden) Markov Models. Apart from the overall accuracy, relevant measures for PoS taggers concern the accuracy of handling unknown words, the amount of training data required (the learning rate), training time, tagging time, and the accuracy on different types of corpora. TnT, a trigram HMM tagger by Brants [1], has shown good results in both accuracy and training time as well as tagging time.

During the development of the CGN corpus, a number of methods for PoS tagging were compared and used for bootstrapping. Zavrel and Daelemans [13] report on a number of PoS taggers trained and tested

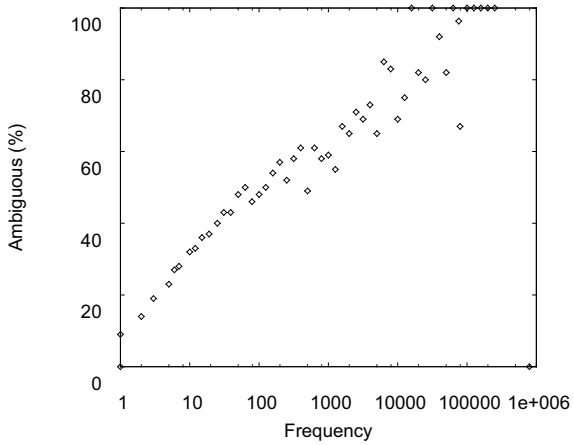


Figure 1: The average percentage of ambiguous words related to the number of occurrences in the CGN corpus.

on the CGN. The best overall performance reported in this study was 95.44%. Canisius and van den Bosch [4] used a subset of the CGN for constructing a memory based PoS tagger with a tagging performance of 95.96%.

The aim of this research is to find the appropriate ingredients for constructing a PoS tagger for spoken Dutch, using Neural Networks. One of the motivations is that, once trained, a Neural Network has efficient tagging performance. It should be remarked that in this study we focus on accuracy and handling of unknown words.

The methodology we use for designing a Neural Network based PoS tagger is as follows. First we determine which input features to use, this selection of input features is based on results found in the literature for different corpora. We will focus explicitly on the coding of unknown words.

Next we determine a small range of (near) optimal window sizes. Afterwards we evaluate different sizes for the hidden layer and select a small set of best ones, based on the performance. Finally we will evaluate the different combinations of the found parameters for window size and number of hidden neurons.

The Corpus Gesproken Nederlands (CGN - Spoken Dutch Corpus) will be introduced in Section 1.1. The detailed design approach for the Neural Network is explained in Section 2. In Section 3 the best found Neural Network will be evaluated. The final conclusions can be found in Section 4.

1.1 Spoken Dutch Corpus (CGN)

The Spoken Dutch Corpus (CGN) [8] is a database of contemporary spoken Dutch. It consists of almost 9 million transcribed words of spoken Dutch, divided into 15 different categories, cf. Table 1. Of these words, around two thirds originate from the Netherlands, the remaining one third from Flanders. The entire CGN is annotated with a large set of PoS tags. The full set consists of 316 different tags, which denote many different features of a word class. An example of such a tag is N(soort,ev,basis,zijd,stan) (noun, sort name, singular, basis (not diminutive), not neuter, standard case (not dative or genitive)). A full explanation of the features and their use can be found in [12]. Many of the pronouns contain even more features, up to nine. This subdivision is so fine-grained that many tags occur only a few times in the entire corpus. There are even 19 tags that occur only once. Although it is possible to discard all the subclasses and use only the main class, this would leave us with a set of only 12 tags (including LET and SPEC, for punctuation mark and special respectively). A tag set of this size is much smaller than what is commonly used in PoS tagging. Discarding all these features also reduces the value of the tagged data to further processing steps. To overcome this problem, the syntactic annotations use a different tag set, consisting of 72 tags. These tags are a reduced version of the full tag set. Only about ten percent of the corpus is tagged using these tags, but the tags can be automatically derived for the rest of the corpus using a set of simplification rules and the full tags. Table 2 shows an overview of this tag set. The challenges of using Neural Networks (NN's) on such a large corpus as the CGN is the size of the training set which requires a separation of the corpus, and to find a workable representation of the input.

Category	Type	Size in words
A	Face to face conversations	2626172
B	Interview with teacher Dutch	565433
C	Phone dialogue (recorded at platform)	1208633
D	Phone dialogue (recorded with mini disc)	853371
E	Business conversations	136461
F	Interviews and discussions recorded from radio	790269
G	Political debates, discussions and meetings	360328
H	Lectures	405409
I	Sport commentaries	208399
J	Discussions on current events	186072
K	News	368153
L	Commentaries on radio and TV	145553
M	Masses and ceremonies	18075
N	Lectures and discourses	140901
O	Text read aloud	903043

Table 1: The 15 different categories of the Dutch Spoken Corpus (CGN).

Tag numbers	Part-of-Speech Tag	Tags in the CGN corpus
1 . . . 8	Noun	N1, N2, . . ., N8
9 . . . 21	Verb	WW1, WW2, . . ., WW13
22	Article	LID
23 . . . 49	Pronoun	VNW1, VNW2, . . ., VNW27
50, 51	Conjunction	VG1, VG2
52	Adverb	BW
53	Interjections	TSW
54 . . . 65	Adjective	ADJ1, ADJ2, . . ., ADJ12
66 . . . 68	Preposition	VZ1, VZ2, VZ3
69, 70	Numeral	TW1, TW2
71	Punctuation	LET
72	Special	SPEC

Table 2: Tags in the medium-sized tag set of size 72. The items in the second column are the main classes and correspond to the reduced tag set of 12 tags.

2 Design of ANN based tagger

In order to design a NN for PoS tagging several issues have to be resolved. First the input features for each word have to be determined and also the window size. In this paper we will focus on NN's with look ahead, that is features of words succeeding the word under consideration are used to classify the word.

2.1 Determining the input features

In conclusion, for every word we use at least the relative frequencies (prior distribution) of PoS tags for that word, of course these relative frequencies are based on the training set. Moreover in the training phase for the words in the window preceding the current word, also the actual PoS tag is used. Of course the test phase the predicted tag is used. For all preliminary tests in order to determine the parameters of the neural network we used set0, set1, set2 and set4 as training set and set10 as validation set. Each set consists of around 100,000 sentences and around 900,000 words. Each training and validation was repeated tree times.

2.1.1 Coding of unknown words

One problem with relative frequencies (prior distribution of tags) based on the training set is the occurrence of so-called unknown words (words that are in the test set but not in the training set). One option to overcome this problem is to use equal priors, meaning that each component of the feature vector gets the value $1/72$, since there are 72 possible tags. We used 10-fold cross validation on the training set to compute the relative frequencies and estimate prior probabilities of unknown words. The results can be found in Table 3. Given

Tag nr.	Tag	Rel. Frequency	Tag nr.	Tag	Rel. Frequency
1	N1	33.0%	15	WW7	2.5%
3	N3	14.0%	54	ADJ1	4.0%
5	N5	10.0%	62	ADJ9	2.4%
9	WW1	2.2%	72	SPEC	19.0%
12	WW4	2.3%			

Table 3: The relative frequency (prior probability) of unknown words based on a 10-fold cross validation. Only the relative frequencies with value at least 2% are listed.

the fact that there are 63 tags with relative frequencies less than 2% we decided not to take into account these tags. Hence every unknown word during testing is coded by the normalized values determined by Table 3. It should be remarked that the tags N3 and N5 do not occur very often in the whole corpus, less than 2%, but they are important tags for unknown words.

2.2 Determining the window size

From the research of Marques and Lopes [7] and Schmid [10] one can conclude that a window size of 3 (tags of) words back and 2 words ahead could be reasonable for the input. We constructed and validated 7 Neural Networks with different window sizes, all with a hidden layer consisting of 50 neurons. The validation results can be found in Table 4. The average performances (over 3 runs) all vary around 96% on the validation set (set10). This set consists of 912,660 words (samples). This leads to an estimate of the 95% confidence interval of $\pm 0.022\%$. Hence we can conclude that a 3×2, a 3×3 and 4×3 ($b \times a$ means b words back and a words ahead) window perform the best.

	Window size						
	2x2	2x3	3x2	3x3	3x4	4x3	4x4
run1	96.52	96.66	96.53	96.64	96.59	96.62	96.52
run2	96.54	96.51	96.64	96.57	96.53	96.56	96.55
run3	96.45	96.61	96.66	96.55	96.56	96.65	96.63
average	**96.503**	**96.593**	**96.610**	**96.587**	**96.560**	**96.610**	**96.567**

Table 4: The performance in % for different window sizes, $b \times a$ means a window size of b tags back and a tags ahead. Each neural network has a hidden layer of 50 neurons.

2.3 Determining the size of the hidden layer

In order to determine the optimal number of hidden neurons we used the smallest window size, 3×2, of the previous subsection. Once again the different networks where trained 3 times on the union of set1, set2, set3 and set4 and validated on set10. The average results can be found in Table 5. From Table 5 we can conclude that the Neural Networks with 250 and 370 hidden neurons are the two best performing ones.

2.4 Determining the optimal configuration

In order to determine the best configuration we combine the results of the previous two subsections, window size 2×3 or 3×2 or 4×3 and number of hidden neurons 250 or 370. This results in 6 different configuration,

	Number of hidden neurons									
	50	90	130	170	210	250	290	330	370	410
Av. Perf.	96.600	96.653	96.697	96.680	96.777	96.787	96.693	96.767	96.803	96.707

Table 5: The performance for different sizes of the hidden layer for an input window of 3×2. Once again the 95% confidence interval is $\pm 0.022\%$.

cf. Table 6, which were compared using the same procedure as described above. Given the results of Table 6

	Window size		
Hidden neurons	2x3	3x2	4x3
250	96.677	96.787	96.743
370	96.663	96.803	**96.830**

Table 6: The average performance of the different configurations over 3 runs. Once again the 95% confidence interval is $\pm 0.022\%$.

we conclude that a Neural Network with a window of 4 words back, 3 words ahead and 370 hidden neurons is the winner of the test. This network will be used for the evaluation.

3 Evaluation

In the evaluation we used set1 up to and including set9 as training set. We will train the Neural Network configuration – window of 4 words back, 3 words ahead and 370 hidden neurons – several times and use set10 for validation. The best performing Neural Network on set10 will be used for testing on set0. Due to the enormous size of the training set, approximately 8 million words, batch learning is not feasible. Hence a different online learning approach is taken. In the first iteration a linear scan through the training set is done. In the following iteration a randomized walk through the training set is used. In every step of this walk a random next line, uniform between 1 and 8, for training is selected. This is repeated 800,000 times in one training epoch. The results on the validation set, set10, are depicted in Table 7. From this table it can be

	Training epoch								
	1	2	3	4	5	6	7	8	9
Performance on set10	97.13	97.25	97.22	**97.36**	97.34	97.33	97.35	97.33	97.33

Table 7: The performance of the Neural Network with a window of 4 words back, 3 words ahead and 370 hidden neurons for the different training epochs.

concluded that the Neural Network after the 4th epoch performed best. This neural network will be used for evaluation of the ultimate test set; set0.

3.1 Overall performance on the test set

The trained Neural Network reached a performance of 97.35% on the test, compared to a performance of 91.86% of the baseline tagger. The baseline tagger assigns to every word the most likely class based on the class probabilities estimated from the data in the training set.

A detailed evaluation of the performance of the Neural Network can be found in Table 8. In this table the performance of the Neural Network on the different categories is compared with the performance of the baseline tagger. From this table it can be concluded that there is a correlation between the size of the category and the performance; the larger the size, the better the performance. Moreover the Neural Network outperforms the baseline tagger with at least 4.5% on each category.

Category	Words	NN	Baseline Tagger
Face to face conversations	82069	97.34%	92.36%
Interviews with teacher Dutch	58307	97.16%	91.65%
Phone dialogue (recorded at platform)	72443	97.93%	93.42%
Phone dialogue (recorded with mini disc)	75706	97.96%	93.31%
Business conversations	13409	97.79%	92.45%
Interview and discussions record from radio	78890	96.96%	90.28%
Political debates and discussions	35341	96.55%	89.52%
Lectures	41032	97.36%	91.40%
Sport comments	21816	97.24%	90.78%
Discussions on current events	18183	97.22%	90.45%
News	35819	96.81%	90.94%
Comments (radio, TV)	13988	96.35%	89.23%
Masses, ceremonies	1762	96.20%	89.10%
Lectures, discourses	13244	96.07%	89.14%
Texts, read aloud	89502	96.31%	89.39%
Average result		**97.35%**	**91.86%**

Table 8: The performance of the trained NN for the different categories of the test set; set0. For comparison the performance results for the baseline tagger are depicted in the last column.

True	Predicted	Errors	Known words	Unknown words
N3	N1	1177	37	1140
N5	N1	823	111	712
Spec	N1	901	109	792
N5	Spec	601	533	68
WW2	WW4	1458	1458	0
WW4	WW2	976	976	0
VZ1	VZ2	857	875	0
VZ2	VZ1	700	700	0

Table 9: The most significant absolute confusions between the different tags and the contributions from the known and unknown words.

Since we have 72 tags, we will not give the total confusion matrix but only the most significant confusions, cf. Table 9. It follows from Table 9 that there is a large confusion between N1, N3, N5 and Spec. Moreover most of these confusions are due to the unknown words. One of the reasons for this confusion for unknown words could be due to the coding of unknown words, cf. Table 3. The most likely tag of an unknown word is N1.

On the other hand the confusions for the other tags are due to the known words. For the known words there is a bilateral confusion between WW2 and WW4. These tags have very low relative frequencies for the unknown words. Hence these tags are almost never the desired tag of an unknown word. The overall performance of the designed PoS tagger on the known words in the test set is 97.88% and most errors are due to the confusion between the tags WW2 – WW4 (contribution of 0.27% to the error) and VZ1 – VZ2 (contribution of 0.17% to the error). On the unknown words the performance of the NN based PoST is 41.67% and the confusion between N3 and N1 contributes 13.29% to the error. The confusion N5 – N1 and Spec – N1 both contribute around 9% to the error. We also tested our approach with equal prior probabilities for the unknown words but this resulted in a performance of 28.51% on the unknown words and a similar performance on known words. Hence our approach of coding unknown words boosted the performance on unknown words by almost 13%.

3.2 Comparison

The following table, Table 10, gives a comparison of different taggers on the CNG corpus[1].

Technique	ALL	Known	Unknown	Comments
NN (this paper)	97.4	97.9	41.7	
SVM [9]	97.5	97.3	70.0	(a)
TnT [11]	97.3	97.5	96.0	(b)
Brill [11]	96.1	97.1	94.4	(c)
Zavrel [13]	95.4			(d)
Canisius [4]	91.9			(d)

Table 10: Performance of different PosT on the CNG

Some comments are in place, the * refer to entries in Table 10:

(a) The SVM tagger uses the same 72 tagset and the same training and test data. Moreover it uses compound analysis for unknown. We expect that NN outperforms SVM if it uses this also. A higher tagging speed makes NN more practical than the SVM tagger.

(b) The TnT tagger uses the same 72 tagset and the same training and test data as the NN and SVM taggers. Build with the TnT tagger of T. Brants, [2].

(c) Uses the Brill tagger, [3]. Same medium sized tag set but trained on 100.000 sentences only. Training on the full training set is not doable.

(d) These results on the CGN corpus are hardly comparable because of different training size and different tag sets.

4 Conclusions

In this paper we designed a Neural Network for Part-of-Speech tagging on Dutch corpora. More specifically we used the Corpus Gesproken Nederlands (CNG) for the design of the Neural Network. The Neural Network uses a sliding window of 4 words back and 3 words ahead. The hidden layer consists of 370 neurons. The input features for the Neural Network are based on a literature study and resulted in a relative frequency coding for the word under consideration and words ahead and the Part-of-Speech tag for the words back.

[1] We are grateful to Herman Stehouwer for his contribution in comparing the results of the NN tagger with those obtained with the TnT and Brill tagger.

Special attention is paid to the coding of unknown words; this is based on the relative frequencies determined by a 10-fold cross validation on the training set.

This design resulted in a performance of 97.35% (Table 8) with a 95% confidence interval of $\pm 0.02\%$. A more detailed analysis showed a performance of 97.88% on known words and 41.67% on unknown words. This performance is comparable to state of the art PoST on the CNG corpus, cf. Section 3.2.

From the analysis of the confusion matrix it followed that large confusions are either totally due to the known words or totally due to the unknown words (Table 9). The developed coding of unknown words based on the relative frequencies for unknown words based on a 10-fold cross validation shows an improvement of almost 13% with respect to a coding with equal priors.

Since many words in Dutch are compounds it is our opinion that still more improvement can be gained by a finer coding on unknown words based on the compound analysis of the unknown word.

References

[1] T. Brants. TnT – A statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP-2000)*, 2000.

[2] Thorsten Brants. Tnt: A statistical part of speech tagger. In *Proceedings of the 6th applied NLP conference, ANLP-2000*, 2000.

[3] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[4] S. Canisius and A. van den Bosch. A memory-based shallow parser for spoken dutch. ILK/Computational Linguistics and AI, Tilburg University, 2004.

[5] Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. Mbt: A memory-based part of speech tagger-generator. In *Proceedings of the 4th Workshop on Very Large Corpora, ACL SIGDAT*, 2000.

[6] T. G. Dietterich. Machine learning for sequential data: A review. In T. Caelli, A. Amin, R. P. W. Duin, M. S. Kamel, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 15–30. Springer, 2002.

[7] N.C. Marques and G.P. Lopes. Tagging with small training corpora. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, volume 2189 of *Lecture Notes in Computer Science*, pages 63 – 72, 2001.

[8] N. Oostdijk, W. Goedertier, F. Van Eynde, L. Boves, J.-P. Martens, M. Moortgat, and H. Baayen. Experiences from the spoken dutch corpus project. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, pages 340 – 347, 2002.

[9] M. Poel, L. Stegeman, and H.J.A. op den Akker. A support vector machine approach to dutch part-of-speech tagging. In M.R. Berthold, J. Shawe-Taylor, and N. Lavrac, editors, *Advances in Intelligent Data Analysis VII. Proceedings of the 7th International Symposium on Intelligent Data Analysis, IDA 2007*, volume 4723 of *Lecture Notes in Computer Science*, pages 274–283. Springer Verlag, 2007.

[10] H. Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the 15th Conference on Computer Linguistics*, volume 1, pages 172 – 176, 1994.

[11] J.H. Stehouwer. Comparing a tbl tagger with an hmm tagger: time efficiency, accuracy, unknown words. Internal report, Dept. Computer Science, University of Twente, 2006.

[12] F. van Eynde. Part of speech tagging en lemmatisering. Technical report, Centrum voor Computerlinguïstiek, K.U. Leuven, 2000.

[13] J. Zavrel and W. Daelemans. Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, 2002.

The Dynamics of Human Behaviour in Poker

Marc Ponsen [a] Karl Tuyls [b] Steven de Jong [a]
Jan Ramon [c] Tom Croonenborghs [d] Kurt Driessens [c]

[a] *Universiteit Maastricht, Netherlands*
[b] *Technische Universiteit Eindhoven, Netherlands*
[c] *Katholieke Universiteit Leuven, Belgium*
[d] *Biosciences and Technology Department, KH Kempen University College, Belgium*

Abstract

In this paper we investigate the evolutionary dynamics of strategic behaviour in the game of poker by means of data gathered from a large number of real-world poker games. We perform this study from an evolutionary game theoretic perspective using the Replicator Dynamics model. We investigate the dynamic properties by studying how players switch between different strategies under different circumstances, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. We illustrate the dynamics using a simplex analysis. Our experimental results confirm existing domain knowledge of the game, namely that certain strategies are clearly inferior while others can be successful given certain game conditions.

1 Introduction

Although the rules of the game of poker are simple, it is a challenging game to master. There exist many books written by domain experts on how to play the game (see, e.g., [2, 4, 9]). A general consensus is that a winning poker strategy should be adaptive: a player should change the style of play to prevent becoming too predictable, but moreover, the player should adapt the game strategy based on the opponents. In the latter case, players may want to vary their actions during a specific game, but they can also consider changing their overall game strategy over a series of games (e.g., play a more aggressive or defensive style of poker).

Although some studies exist on modeling poker players and providing a best-response given the opponent model (see, e.g., [1, 8, 10]), not much research focuses on overall strategy selection. In this paper we address this issue by investigating the evolutionary dynamics of strategic player behaviour in the game of poker. We perform this study from an evolutionary game-theoretic perspective using the Replicator Dynamics (RD) [5, 6, 11, 12]. More precisely, we investigate the dynamic properties by studying how players switch between different strategies (based on the principle of selection of the fittest), under different circumstances, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are.

A complicating factor is that the RD can only be applied straightforwardly to simple normal form games as for instance the Prisoner's Dilemma game [3]. Applying the RD to poker by assembling the different actions in the different phases of the game for each player will not work, because this leads to an overly complex table with too many dimensions. To address this problem, overall strategies (i.e., behaviour over a series of games, henceforth referred to as *meta strategies*) of players may be considered. Using these meta strategies, a heuristic payoff table can then be created that enables us to apply different RD models and perform our analysis. This approach has been used before in the analysis of behaviour of buyers and sellers in automated auctions [7, 13, 14]. Conveniently, for the game of poker several meta strategies are already defined in literature. This allows us to apply RD to the game of poker. An important difference with previous work, is that we use real-world poker games from which the heuristic payoff table is derived, as opposed to the artificial data used in the auction studies. We observed poker games played on a poker website, in which human players competed for real money at various stakes.

Therefore, the contributions of this paper are twofold. First, we provide new insights in the dynamics of strategic behaviour in the complex game of poker using RD models. These insights may prove useful for strategy selection by human players but can also aid in creating strong artificial poker players. Second, unlike other studies, we apply RD models to real-world human data.

The remainder of this paper is structured as follows. We start by explaining the poker variant we focus on in our research, namely No-Limit Texas Hold'em poker, and describe some well-known meta strategies for this game. Next we elaborate on the Replicator Dynamics and continue with a description of our methodology. We end with experiments and a conclusion.

2 Background

In this section we will first briefly explain the rules of the game of poker. Then we will discuss meta strategies as defined by domain experts.

2.1 Poker

Poker is a card game played between at least two players. In a nutshell, the object of the game is to win games (and consequently win money) by either having the best card combination at the end of the game, or by being the only active player. The game includes several betting rounds wherein players are allowed to invest money. Players can remain active by at least matching the largest investment made by any of the players, or they can choose to fold (i.e., stop investing money and forfeit the game). In the case that only one active player remains, i.e., all other players chose to fold, the active player automatically wins the game. The winner receives the money invested by all the players.

In this paper we focus on the most popular poker variant, namely No-Limit Texas Hold'em. This game includes 4 betting rounds (or phases), respectively called the pre-flop, flop, turn and river phase. During the first betting round, all players are dealt two private cards (what we will now refer to as a player's *hand*) that are only known to that specific player. To encourage betting, two players are obliged to invest a small amount the first round (the so-called small- and big-blind). One by one, the players can decide whether or not they want to participate in this game. If they indeed want to participate, they have to invest at least the current bet. This is known as *calling*. Players may also decide to *raise* the bet. If they do not wish to participate, players *fold*, resulting in possible loss of money they bet thus far. During the remaining three betting phases, the same procedure is followed. In every phase, community cards appear on the table (respectively 3 in the flop phase, and 1 in the other phases). These cards apply to all the players and are used to determine the card combinations (e.g., a pair or three-of-a-kind may be formed from the player's private cards and the community cards).

2.2 Meta strategies

There exists a lot of literature on winning poker strategies, mostly written by domain experts (see, e.g., [2, 4, 9]). These poker strategies may describe how to best react in detailed situations in a poker game, but also how to behave over large numbers of games. Typically, experts describe these so-called meta strategies based on only a few features. For example, an important feature in describing a player's meta strategy is the percentage of times this player voluntarily sees the flop (henceforth abbreviated as VSF), since this may give insight in the player's hand selection. If a particular player chooses to play more than, let's say, 40% of the games, he or she may play with less quality hands (see [9] for hand categorization) compared to players that only see the flop rarely. The standard terminology used for respectively the first approach is a *loose* and for the latter a *tight* strategy. Another important feature is the so-called *aggression-factor* of a player (henceforth abbreviated as AGR). The aggression-factor illustrates whether a player plays offensively (i.e., bets and raises often), or defensively (i.e., calls often). This aggression factor is calculated as:

$$\frac{\%\text{bet} + \%\text{raise}}{\%\text{calls}}$$

A player with a low aggression-factor is called *passive*, while a player with a high aggression-factor is simply called *aggressive*. The thresholds for these features can vary depending on the game context. Taking into

account these two features, we can construct four meta strategies, namely: 1) loose-passive (LP), 2) loose-aggressive (LA), 3) tight-passive (TP), and 4) tight-aggressive (TA). Again note that these meta-strategies are derived from poker literature.

Experts argue that the TA strategy is the most profitable strategy, since it combines patience (waiting for quality hands) with aggression after the flop. One could already claim that any aggressive strategy dominates all passive strategies, simply by looking at the rules of the poker game. Note that games can be won by having the best card combination, but also by betting all opponents out of the pot. However, most poker literature will argue that adapting a playing style is the most important feature of any winning poker strategy. This applies to detailed poker situations, i.e., varying actions based on current opponent(s), but also varying playing style on a broader scale (e.g., switching from meta strategy). We will next investigate how players (should) switch between meta strategies in the game of No-Limit Texas Hold'em poker.

3 Methodology

In this section we concisely explain the methodology we will follow to perform our analysis. We start by explaining Replicator Dynamics (RD) and the heuristic payoff table that is used to derive average payoffs for the various meta strategies. Then we explain how we approximate the Nash equilibria of interactions between the various meta strategies. Finally, we elucidate our algorithm for visualizing and analyzing the dynamics of the different meta strategies in a simplex plot.

3.1 Replicator Dynamics

The RD [11, 16] are a system of differential equations describing how a population of strategies evolves through time. The RD presumes a number of agents (i.e., individuals) in a population, where each agent is programmed to play a pure strategy. Hence, we obtain a certain mixed population state x, where x_i denotes the population share of agents playing strategy i. Each time step, the population shares for all strategies are changed based on the population state and the rewards in a payoff table. Note that single actions are typically considered in this context, but in our study we look at meta strategies.

An abstraction of an evolutionary process usually combines two basic elements, i.e., selection and mutation. Selection favors some population strategies over others, while mutation provides variety in the population. In this research, we will limit our analysis to the basic RD model based solely on selection of the most fit strategies in a population. Equation 1 represents this form of RD.

$$\frac{dx_i}{dt} = [(Ax)_i - x \cdot Ax]\, x_i \tag{1}$$

In Equation 1, the state x of the population can be described as a probability vector $x = (x_1, x_2, ..., x_n)$ which expresses the different densities of all the different types of replicators (i.e., strategies) in the population, with x_i representing the density of replicator i. A is the payoff matrix that describes the different payoff values that each individual replicator receives when interacting with other replicators in the population. Hence $(Ax)_i$ is the payoff that replicator i receives in a population with state x, whereas $x \cdot Ax$ describes the average payoff in the population. The growth rate $\frac{dx_i}{dt}/x_i$ of the proportion of replicator i in the population equals the difference between the replicator's current payoff and the average payoff in the population. For more information, we refer to [3, 5, 15].

3.2 The Heuristic Payoff Table

The heuristic payoff table represents the payoff table of the poker game for the different meta strategies the different agents can employ. In essence it replaces the Normal Form Game (NFG) payoff table for the atomic actions. For a complex game such as poker it is impossible to use the atomic NFG, simply because the table has too many dimensions to be able to represent it. Therefore, we look at heuristic strategies as outlined in Section 2.2.

Let's assume we have A agents and S strategies. This would require S^A entries in our NFG table. We now make a few simplifications, i.e., we do not consider different types of agents, we assume all agents can choose from the same strategy set and all agents receive the same payoff for being in the same situation. This setting corresponds to the setting of a symmetric game. This means we consider a game where the payoffs for playing a particular strategy depend only on the strategies employed by the other agents, but not on who

is playing them. Under this assumption we can seriously reduce the number of entries in the heuristic payoff table. More precisely, we need to consider the different ways of dividing our A agents over all possible S strategies. This boils down to:

$$\binom{A+S-1}{A}$$

Suppose we consider 3 heuristic strategies and 6 agents, this leads to a payoff table of 28 entries, which is a serious reduction from $3^6 = 729$ entries in the general case. As an example the next table illustrates what the heuristic payoff table looks like for three strategies S_1, S_2 and S_3.

$$P = \left(\begin{array}{ccc|ccc} S_1 & S_2 & S_3 & U_1 & U_2 & U_3 \\ \hline s_1 & s_2 & s_3 & u_1 & u_2 & u_3 \\ & ... & & & ... & \end{array}\right)$$

Consider for instance the first row of this table: in this row there are s_1 agents that play strategy S_1, s_2 agents that play strategy S_2 and s_3 agents play strategy S_3. Furthermore, u_i is the respective expected payoff for playing strategy S_i. We call a tuple $(s_1, s_2, s_3, u_1, u_2, u_3)$ a profile of the game. To determine the payoffs u_i in the table, we compute expected payoffs for each profile from real-world poker data we assembled. More precisely, we look in the data for the appearance of each profile and compute from these data points the expected payoff for the used strategies. However, because payoff in the game of poker is non-deterministic, we need a significant number of independent games to be able to compute representative values for our table entries. In Section 4 we provide more details on the data we used and on the process of computing the payoff table.

3.3 Approximating Nash Equilibria

In this section we describe how we can determine which of our restpoints of the RD are effectively Nash equilibria (so note that a restpoint of the RD is not necessarily Nash). The approach we describe is based on work of Walsh et al. and Vytelyngum et al. [13, 14]. An Nash equilibria occurs when no player can increase its payoff by changing strategy unilaterally. For the sake of clarity we follow the notation of [14].

The expected payoff of an agent playing a strategy $j \in S$ [1], given a mixed-strategy p (the population state), is denoted as $u(e_j, p)$. This corresponds to $(Ax)_i$ in Equation 1. The value of $u(e_j, p)$ can be computed by considering the results from a large number of poker games with a player playing strategy j and the other agents selected from the population, with a mixed-strategy p. For each game and every strategy, the individual payoffs of agents using strategy j are averaged. The Nash equilibrium is then approximated as the argument to the minimisation problem given in Equations 2 and 3.

$$v(p) = \sum_{j=1}^{S} (max[u(e_j, p) - u(p, p), 0])^2 \tag{2}$$

$$p_{nash} = argmin_p[v(p)] \tag{3}$$

Here, $u(p, p)$ is the average payoff of the entire population and corresponds with term $x \cdot Ax$ of Equation 1. Specifically, p_{nash} is a Nash equilibrium if and only if it is a global minimum of $v(p)$, and p is a global minimum if $v(p) = 0$. We solve this non-linear minimisation problem using the Amoeba non-linear optimiser [14].

3.4 Simplex Analysis

The simplex analysis allows us to graphically and analytically study the dynamics of strategy changes. Before explaining this analysis, we first introduce a definition of a simplex. Given n elements which are randomly chosen with probabilities $(x_1, x_2, \ldots, x_n)$, there holds $x_1, x_2, \ldots, x_n \geq 0$ and $\sum_{i=1}^{n} x_i = 1$. We denote the set of all such probability distributions over n elements as Σ_n or simply Σ if there is no confusion possible. Σ_n is a $(n-1)$-dimensional structure and is called a *simplex*. One degree of freedom is lost due to the normality constraint. For example in Figure 1, Σ_2 and Σ_3 are shown. In the figures throughout the experiments we use Σ_3, projected as an equilateral triangle as in Figure 1(b), but we drop the axes and

[1] The use of S differs from that in Section 3.2. Here S represents the set of strategies, unlike the number of strategies in Section 3.2.

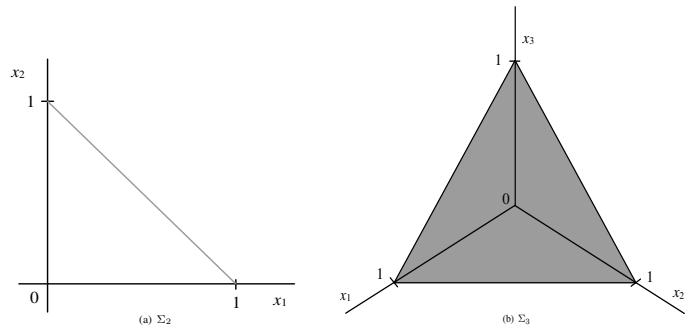


Figure 1: The unit simplices Σ_2 (a; left) and Σ_3 (b; right).

labels. Since we use four meta strategies and Σ_3 concerns only three, this implies that we need to show four simplexes Σ_3, from each of which one strategy is missing.

Using the generated heuristic payoff table, we can now visualize the dynamics of the different agents in a simplex as follows. To calculate the RD at any point $s = (x_1, x_2, x_3)$ in our simplex, we consider N (i.e., many) runs with mixed-strategy s; x_1 is the percentage of the population playing strategy S_1, x_2 is the percentage playing strategy S_2 and x_3 is is the percentage playing strategy S_3. For each run, each poker agent selects their (pure) strategy based on this mixed-strategy. Given the number of players using the different strategies (S_1, S_2, S_3), we have a particular profile for each run. This profile can be looked up in our table, yielding a specific payoff for each player. The average of the payoffs of each of these N profiles gives the payoffs at $s = (x_1, x_2, x_3)$. Provided with these payoffs we can easily compute the RD by filling in the values of the different variables in Equation 1. This yields us a gradient at the point $s = (x_1, x_2, x_3)$.

Starting from a particular point within the simplex, we can now generate a smooth trajectory (consisting of a piecewise linear curve) by moving a small distance in the calculated direction, until the trajectory reaches an equilibrium. A trajectory does not necessarily settle at a fixed point. More precisely, an equilibrium to which trajectories converge and settle is known as an attractor, while a saddle point is an unstable equilibrium at which trajectories do not settle. Attractors and saddle points are very useful measures of how likely it is that a population converges to a specific equilibrium.

4 Experiments and results

We collected a total of 1599057 No-Limit Texas Hold'em games with 6 or more players starting. As a first step we needed to determine the strategy for a player at any given point. If a player played less than 50 games in total, we argue that we do not have sufficient data to establish a strategy, and therefore we ignore this player (and game). If the player played at least 50 games, we used an interval of 50 games to collect statistics for this specific player, and then determined the VSF and AGR values. We set the thresholds respectively to 0.35 and 2.0, i.e., if $VSF > 0.35$, then the player is considered loose (and tight otherwise), and if $AGR > 2$ then the player is considered aggressive (and passive otherwise). These are commonly used thresholds for a No-Limit Texas Hold'em game (see e.g., [2, 4, 9]). The resulting strategy was then associated with the specific player for all games in the interval of 50 games. Having estimated all players' strategies, it is now possible to determine the table configuration (i.e., the number of players playing any of the four meta strategies) for all games. Finally, we can compute the average payoffs for all strategies given a particular table configuration and produce a profile (see Section 3.2).

We plotted four simplexes that resulted from our RD analysis in Figure 2. Recall from Section 3.4 that these simplexes show the dynamic behavior of the participating players having a choice from three strategies. This means that the evolution of the strategies, employed in the population, is visualized for every possible initial condition of the game. The initial condition determines in which basin of attraction we end up, leading to some specific attractor or repeller. These restpoints (i.e. attractors or repellers) are potentially Nash equilibria.

What we can immediately see from the plots is that both passive strategies LP and TP (except in plot a) are repellers. In particular the LP strategy is a strong repeller. This suggests that no matter what the game situation is, when playing the LP strategy, it is always rational to switch strategy to for example TA or LA.

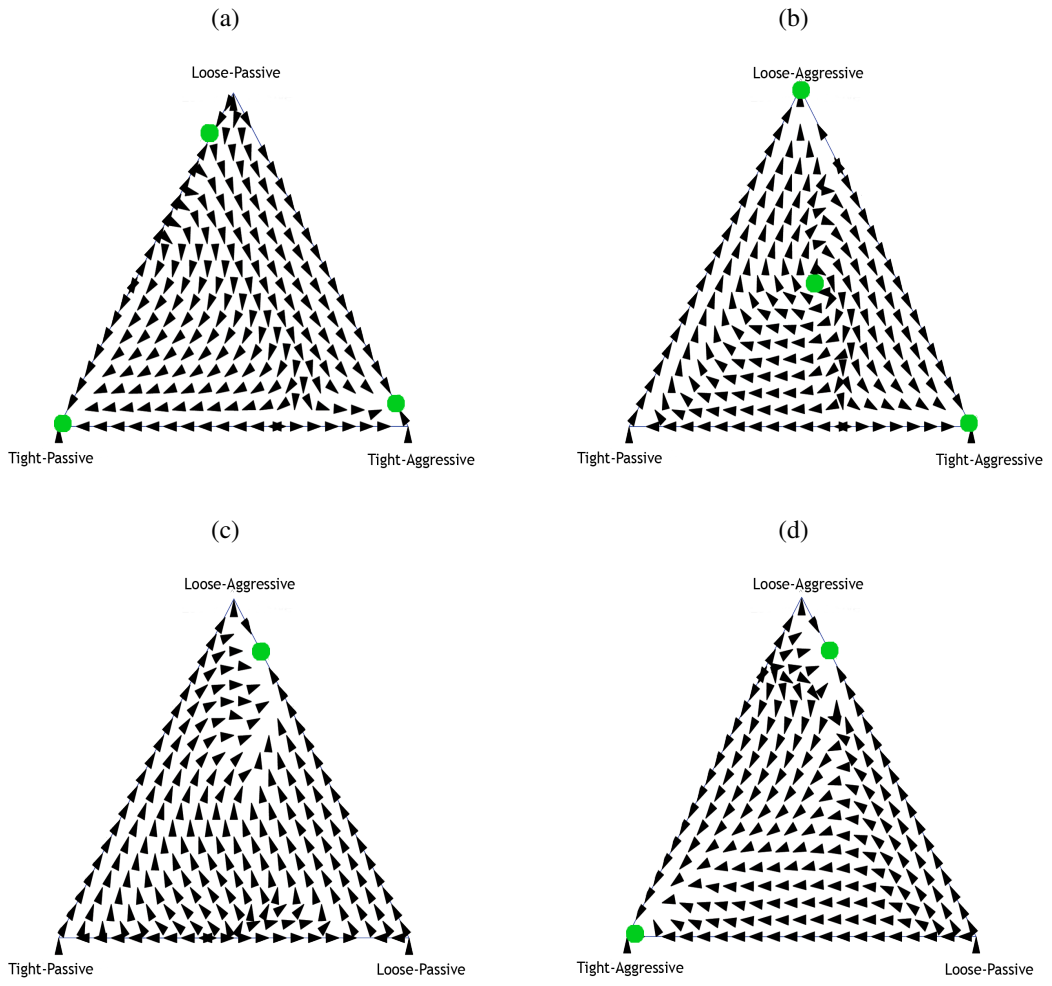


Figure 2: The direction field of the RD using the heuristic payoff table considering the four described meta-strategies. Dots represent the Nash equilbria.

This nicely confirms the claim made earlier (and in literature), namely that aggressive strategies dominate their passive counterparts.

The dots indicated on the plots represent the Nash equilibria of the respective games[2]. Figure 2a contains three Nash equilibria of which two are mixed and one is pure. The mixed equilibrium at the axis TP-LP is evolutionarily unstable as a small deviation in a players' strategy might lead the dynamics away from this equilibrium to one of the others. The mixed equilibrium at the axis LP-TA is stable. As one can see this equilibrium lies close to the pure strategy TA. This means that TA is played with a higher probability than LP. Finally, there is also one stable pure equilibrium present, i.e., TP. Of the stable equilibria TP has the largest basin of attraction.

Figure 2b contains 3 Nash equilibria of which one is mixed and two are pure. As one can see from the picture, the mixed Nash equilibrium is evolutionarily unstable, i.e., any small perturbation of this equilibrium immediately leads the dynamics away from it to one of the other pure Nash equilibria. This means that if one of the players would decide to slightly change its strategy at the equilibrium point, the dynamics of the entire population would drastically change. The mixed Nash equilibrium almost corresponds to the situation in which the three strategies are played with equal probability, i.e., a uniform distribution. The pure Nash equilibria LA and TA are both evolutionarily stable. LA has a larger basin of attraction than TA (similar to plot a), which does not completely correspond with the expectations of domain experts (it is assumed by

[2]Due to space constraints we only discuss the Nash equilibria of Figures 2a-2b and Figures 3a-3b. For completeness the equilibria of Figures 2c and 2d are also indicated.

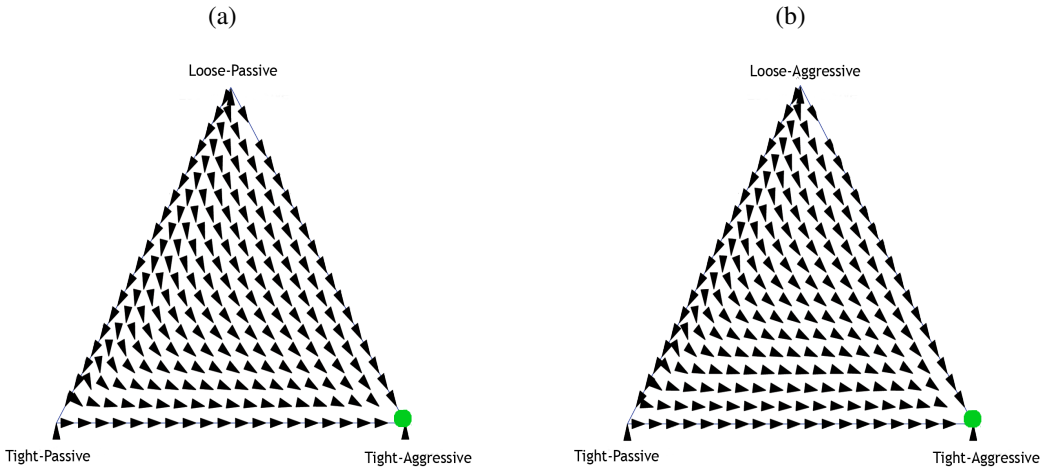


Figure 3: The direction field of the RD using the heuristic payoff table using data of games with active players at the flop.

domain experts that in general TA is the most profitable strategy).

One possible explanation is the following: we noticed that some strategies (depending on the used thresholds for VSF and AGR) are less played by humans compared to other strategies. Therefore, a table configuration with a large number of agents playing these scarcely played strategies, results in few instances and possibly a distorted average payoff due to the high variance of profits in the game of No-Limit Texas Hold'em. In particular, we observed that table configurations with many humans playing a tight strategy had only few instances (e.g., the payoffs used in plot a, with two tight strategies in the simplex, were calculated using 40% less instances compared to those in plot b). A severe constraint on the number of instances is currently our chosen representation for a profile. In the previous experiment, we used games with 6 or more starting players, and counted the number of occurrences of the four strategies. An alternative way of interpreting the data is only considering players active at the flop. Since most of the times only 4 or less players (and a maximum of 6 players in our data) are active at the flop, this results in fewer profiles. Basically, we generalize over the number of players starting at the beginning of the game and only focus on the interaction between strategies during the phases that most influence the average payoffs. The results from these experiments are illustrated in Figure 3.

In Figure 3a and 3b we have one pure Nash equilibrium being a dominant strategy, i.e., TA. These equilibria, and the evolution to them from any arbitrary initial condition, confirm the conclusions of domain experts.

5 Conclusion

In this paper we investigated the evolutionary dynamics of strategic behaviour of players in the game of No-Limit Texas Hold'em poker. We performed this study from an evolutionary game theoretic perspective using Replicator Dynamic models. We investigated the dynamic properties by studying how human players should switch between different strategies under different circumstances, and what the Nash equilibria look like. We observed poker games played at an online poker site and used this data for our analysis. Based on domain knowledge, we identified four distinct meta strategies in the game of poker. We then computed the heuristic payoff table to which we applied the Replicator Dynamic model. The resulting plots confirm that what is claimed by domain experts, namely that often aggressive strategies dominate their passive counterparts, and that the Loose-Passive strategy is an inferior one.

For future work, we will examine the interactions between the meta strategies among several other dimensions, namely, more detailed meta strategies (i.e., based on more features), a varying number of players, different parameter settings and different Replicator Dynamic models (e.g., including mutation). We are also interested in performing this study using simulated data (which we can generate much faster). Finally, since it is clear from our current experiments that the Loose-Passive strategy is an inferior one, we can focus

on the switching dynamics between the remaining strategies given the presence of a fixed number of players playing the Loose-Passive strategy. This way, we focus on the dynamics for the strategies that matter.

6 Acknowledgments

Marc Ponsen is sponsored by the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. Jan Ramon and Kurt Driessens are post-doctoral fellow of the Research Foundation - Flanders (FWO). The authors wish to express their gratitude to P. Vytelingum for his insightful comments on the construction of the heurisitic payoff table.

References

[1] A. Davidson, D. Billings, J. Schaeffer, and D. Szafron. Improved opponent modeling in poker. In *Proceedings of The 2000 International Conference on Artificial Intelligence (ICAI'2000)*, pages 1467–1473, 2000.

[2] D. Doyle Brunson. *Doyle Brunson's Super System: A Course in Power Poker*. Cardoza, 1979.

[3] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2001.

[4] D. Harrington. *Harrington on Hold'em Expert Strategy for No Limit Tournaments*. Two Plus Two Publisher, 2004.

[5] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[6] J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[7] S. Phelps, S. Parsons, and P. McBurney. Automated trading agents versus virtual humans: an evolutionary game-theoretic comparison of two double-auction market designs. In *Proceedings of the 6th Workshop on Agent-Mediated Electronic Commerce*, New York, NY, 2004.

[8] M. Ponsen, J. Ramon, T. Croonenborghs, K. Driessens, and K. Tuyls. Bayes-relational learning of opponent models from incomplete information in no-limit poker. In *Twenty-third Conference of the Association for the Advancement of Artificial Intelligence (AAAI-08)*, pages 1485–1487, Chicago, USA, 2008.

[9] D. Slansky. *The Theory of Poker*. Two Plus Two Publisher, 1987.

[10] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI '05)*, pages 550–558, 2005.

[11] P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Math. Biosci.*, 40:145–156, 1978.

[12] K. Tuyls, P. 't Hoen, and B. Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *The Journal of Autonomous Agents and Multi-Agent Systems*, 12:115–153, 2006.

[13] P. Vytelingum, D. Cliff, and N. R. Jennings. Analysing buyers and sellers strategic interactions in marketplaces: an evolutionary game theoretic approach. In *Proc. 9th Int. Workshop on Agent-Mediated Electronic Commerce*, Hawaii, USA, 2007.

[14] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In P. Gymtrasiwicz and S. Parsons, editors, *Proceedings of the 4th Workshop on Game Theoretic and Decision Theoretic Agents*, 2001.

[15] J. W. Weibull. *Evolutionary Game Theory.* MIT Press, 1996.

[16] E. Zeeman. Dynamics of the evolution of animal conflicts. *Journal of Theoretical Biology*, 89:249–270, 1981.

Creating a Bird-Eye View Map using an Omnidirectional Camera

Steven Roebert Tijn Schmits Arnoud Visser

Intelligent System Laboratory Amsterdam [1]
Universiteit van Amsterdam, 1098 SJ Amsterdam, the Netherlands

Abstract

An omnidirectional camera has properties which are highly beneficial for navigation of a mobile robot. In this paper this is demonstrated with a new application; a visual map of the environment created from bird-eye view perspective. This visual map can be created on-line, which allows an operator to navigate the robot through an unknown environment based on this visual map. The quality of the map is tested in a self-localization experiment.

1 Introduction

Robots operating in an unknown environment can strongly benefit from visual data obtained from their surroundings. Visual information can be obtained with a variety of imaging devices, lenses and accessories. Omnidirectional vision, providing a 360° view of the sensor's surroundings, is popular in the robotics research area (e.g. [5]).

Omnidirectional views can be obtained using multiple cameras, a single rotating camera, a fish-eye lens and or a convex mirror. A catadioptric vision system, consisting of a conventional camera in front of a convex mirror with the center of the mirror aligned with the optical axis of the camera, is the most generally applied technique for omnidirectional vision. Mirrors which are conic, spherical, parabolic or hyperbolic all are able to provide omnidirectional images [1].

An omnidirectional catadioptric camera has some great advantages over conventional cameras. One of them being the fact that visual landmarks remain in the field of view much longer than with a conventional camera, as demonstrated in section 3.3. Also does the imaging geometry have various properties that can be exploited for navigation. A view in all directions on the close surroundings allows to avoid nearby obstacles, allowing navigation through narrow passages and a safe retreat when the robot is cornered.

Omnidirectional vision has played a major part in past and present research. At the Intelligent Systems Laboratory Amsterdam omnidirectional vision is used for Trajectory SLAM and for appearance-based self-localization using a probabilistic framework [3, 2]. For a complete overview of application of omnidirectional vision in robotics, see [7, 5].

At the RoboCup Rescue Simulation League one of the simulators which is used and developed is USARSim [4], the 3-D simulation environment of Urban Search And Rescue (USAR) robots and environments, built on top of the Unreal Tournament game and intended as a research tool for the study of human-robot interfacing and multi-robot coordination [8]. The omnidirectional vision sensor is a recent extension [11] of the USARSim environment. The Virtual Robots Competition, using USARSim as the simulation platform, aims to be the meeting point between researchers involved in the Agents Competition and those active in the RoboCup Rescue League. As the omnidirectional catadioptric camera (see Fig. 1) is an important sensor in the robotics field, this USARSim extension allows new applications. In this paper we will describe such a new application, the creation of a bird-eye view map which reconstructs the texture of the unknown environment.

[1] A part of the research reported here is performed in the context of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

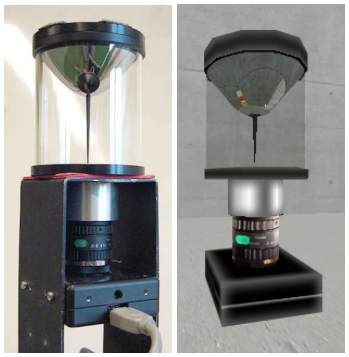


Figure 1: The catadioptric omnidirectional camera, real and simulated.

2 Method

2.1 Creating a Bird-Eye View Image

Omnidirectional image data provided by a parabolic catadioptric omnidirectional camera is hardly intuitive to a human observer (see Fig. 3) as the parts of the image data shows the robot's surroundings upside down, which could makes recognition more complicated, and due to reflection the environment is mirrored. The image also shows heavy counter-intuitive perspective distortion. It has already been shown [6] that image post-processing can greatly improve the readability of omnidirectional data to a human observer. Further, a large amount of automated vision techniques assume perspective projection, which makes the transformation of omnidirectional image data to geometrically correct perspective images highly desirable. Here is described how to transform real omnidirectional image data into the perspective of an observer above the robot: the bird-eye view. This view allows to present the observation in a convenient way to a human operator.

2.1.1 3D to Omnidirectional Pixel Relation

Bird-eye views are obtained by radial correction around the image center. The bird-eye view is a scaled perspective projection of the ground plane, and significantly simplifies the navigation system. For example, corridors appear as image bands of constant width.

Shree K. Nayar describes a direct relation between a location in a 3D environment and the location in the omnidirectional image where this point can be seen if nothing obstructs the view [10]. The correspondence between a pixel in the omnidirectional image $p_{omni} = (x_{omn}, y_{omn})$ and a pixel in the birds-eye view image $p_{be} = (x_{be}, y_{be})$ is defined by the following equations:

$$\theta = \arccos \frac{z}{\sqrt{x_{be}^2 + y_{be}^2 + z^2}}, \quad \phi = \arctan \frac{y_{be}}{x_{be}} \tag{1}$$

$$\rho = \frac{h}{1 + \cos\theta} \tag{2}$$

$$x_{omn} = \rho \sin\theta \cos\phi, \quad y_{omn} = \rho \sin\theta \sin\phi \tag{3}$$

where h is the radius of the circle describing the 90° incidence angle on the omnidirectional camera effective viewpoint. The variable z is defined by the distance between the effective viewpoint and the projection plane in pixels. These equations can be used to construct perspectively correct images based on omnidirectional camera data by translating 3D projection plane pixel locations to omnidirectional pixel locations. The results of these transformations can be seen in Figure 4.

2.1.2 3D to Standard Perspective Pixel Relation

Bird-eye views can also be obtained using data obtained from a standard perspective camera. The correspondence between a pixel in the standard perspective image $p_p = (x_p, y_p)$ and a pixel in the birds-eye view

image $p_{be} = (x_{be}, y_{be})$ is defined by rotation matrices

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-x_\theta) & \sin(-x_\theta) \\ 0 & -\sin(-x_\theta) & \cos(-x_\theta) \end{bmatrix}$$
$$R_y = \begin{bmatrix} \cos(-y_\theta) & 0 & \cos(-y_\theta) \\ 0 & 1 & 0 \\ \cos(-y_\theta) & 0 & \cos(-y_\theta) \end{bmatrix} \quad (4)$$
$$R_z = \begin{bmatrix} \cos(-z_\theta) & \sin(-z_\theta) & 0 \\ -\sin(-z_\theta) & \cos(-z_\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\theta = (x_\theta, y_\theta, z_\theta)$ is defined by the camera orientation, and the equations

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_x R_y R_z \left(\begin{bmatrix} x_{be} \\ y_{be} \\ 0 \end{bmatrix} - \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \right) \quad (5)$$

$$x_p = c\frac{x'}{z'}, \quad y_p = c\frac{y'}{z'} \quad (6)$$

where $l_{cam} = (x_{cam}, y_{cam}, z_{cam})$ is defined by the camera location and c is a scaling factor which can be derived from the camera image width in pixels, the camera FOV angle and the desired resolution of the bird-eye view.

2.2 Creating a Map

To create a map of the environment using the bird-eye view images, it is necessary to know the exact location where the image is taken by the robot. An estimate of the robot's location can be provided with SLAM algorithm, which processes the observations. The observations can be visual (e.g. landmark detection) or based on the accurate laser scanners used. The center of the bird-eye view image would then be placed at the estimated location and from there it is possible to start constructing a map.

2.2.1 Transforming the Images

When the images are taken by the camera, the robot can be at any orientation on the map. Although these images have a field of view of 360° the images will still be different if the robot is facing another direction. This means the images have to be rotated in order to fit on the map. To obtain the required images which are all aligned in the same way, one can simply rotate the images using the same angle as the robot's current rotation on the map. An alternative approach would be to estimate the relative orientation between two omnidirectional images, as done by [6].

As mentioned earlier, the image taken from an omnidirectional camera is mirrored due to reflection. Since we do not want any mirrored images on a map, the image needs to be flipped. After all these transformations it is time to use the transformed image to create the map.

2.2.2 Gradual Overlap

If the acquired images are drawn on the map directly (e.g. the pixels on the map are set to the exact same values of the acquired image), without taken into account the previously drawn images, the major parts of the visual map would be overwritten by new information (dependent on the amount of overlap). Furthermore, small errors in the location of images (e.g. the robot's estimated location was not precise) would be directly visible on the map, and do not smooth out by averaging. To overcome these errors, the image will not be drawn directly but the colors of the map will gradually be changed towards the values of the bird-eye view images. This gradual change can be controlled by the use of the alpha component of the RGBA color space. This alpha component will be a number between 0 and 255 (i.e. possible color values) and will represent the maximum change in color value for a pixel when a camera image is being processed. It will then take multiple images from the camera to build up a visual map. Small errors will be gracefully smooth out as can be seen in Figure 2(a).

2.2.3 Loss of Resolution

A main disadvantage of catadioptric omnidirectional cameras is the loss of resolution. This loss of resolution mainly occurs for pixels further from the center of the image. Pixels further from the center should have less effect on the visual map. This effect can be implemented by adding a 2-dimensional Gaussian filter (equation 7).

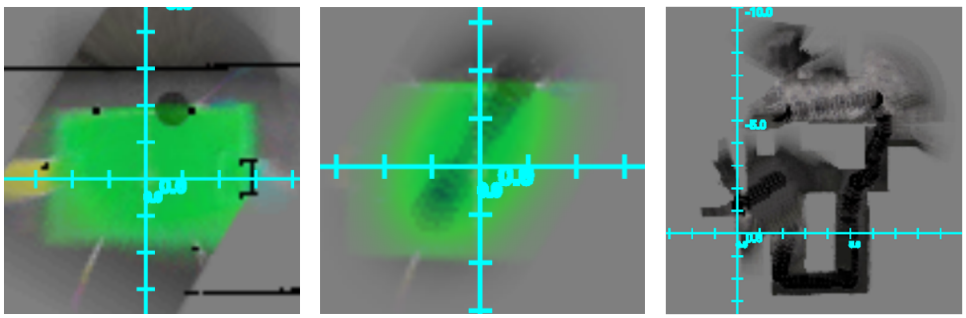
$$e^{\frac{-(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \tag{7}$$

This filter also takes into account the fact that the localization and perspective errors are more likely to occur further from the center of the image. The value of the Gaussian filter is used to scale the alpha component of section 2.2.2. The further away from the center, the lower the value for the Gaussian will be and thus the lower the change in color value will be performed. Because of this filter pixels will change to acquired color more quickly if they lie closer to the center of the image. The result of this filter can be seen in Figure 2(b).

2.2.4 Obstacle Filtering

When creating a bird-eye view from an omnidirectional camera image, there will be one significant difference from an image actually taken directly above the scene. This difference comes from the fact that an omnidirectional camera at the height of the robot can only see the sides of larger objects and not the actual top. So any part of the map that lies behind such an obstacle, will not be visible from the robots point of view. Instead the obstacle will be stretched out over the part that lies behind it. This results in obstacles distorting the visual map behind them, polluting the map from all directions that those obstacles have been seen.

The best way to deal with these objects is to not draw them at all, by applying an additional rendering filter. However, for such filter a form of obstacle detection sensors on the robot should be present. In these experiments laser scanners where used for self localization and object detection. Using the laser scanners one can establish the location of the obstacles as objects in the bird-eye view image. The algorithm to filter the objects is build into the algorithm for merging the camera images with the map (i.e. the algorithm using the alpha component and the Gaussian filter). The algorithm will go through all pixels from the camera image and decides if there exists an object on this location. This is done by consulting the data from the laser scanner that pointed towards the angle of that specific location. If the data provides the evidence for an existing object, every pixel that lies behind this object (in a line from the center of the image) will be filtered. Since we can not know exactly how tall an object will be, we do not consider any pixels behind it. Once applied, this obstacle filter helps creating in a better outline of the visual map as can be seen in Figure 2(c).



(a) Gradual Overlap (b) Gaussian Filter (c) Obstacle Filtering

Figure 2: Three filters used for the creation of the visual map.

2.3 Localization and Mapping

Besides creating a map, we studied the effect of the bird-eye view transformation on the quality of the maps. To estimate the quality, a visual self-localization algorithm was applied on both the omnidirectional

and bird-eye view images. The self-localization algorithm is based on an Extended Kalman Filter (EKF), a recursive filter which estimates a Markov process based on a Gaussian noisy model, is based on formulas provided by Thrun and Burgard in [12]. The robot position μ_t is estimated on the control input u_t and landmark measurement m_t, as defined by Equations 8 to 10 respectively.

$$\mu_t = (x_t, y_t, \phi_t) \tag{8}$$
$$u_t = (v_t, \delta\phi_t) \tag{9}$$
$$m_t = (r_t, \gamma_t) \tag{10}$$

where x_t and y_t define the robot location in world coordinates, ϕ_t defines the robot direction, v_t defines the robot velocity, $\delta\phi_t$ defines robot rotation angle, γ_t defines the landmark perception angle. User input u_t is defined by noisy odometry sensor data and the measured distance to a landmark, r_t, is calculated based on the inverse of Equation 2.

3 Experimental Results

3.1 Image Transformation

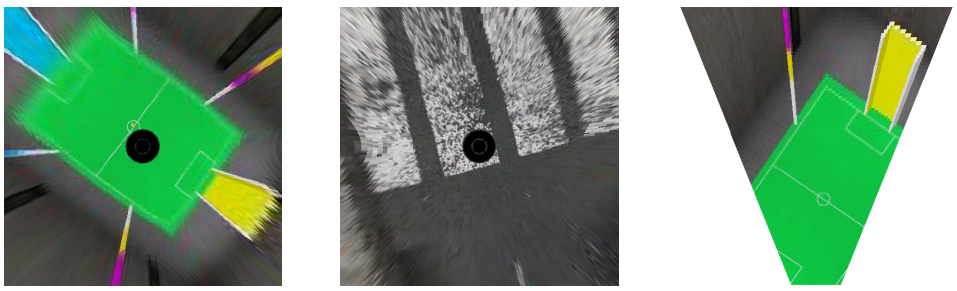
Using the equations 1 to 6, the following images were created. These resulting bird-eye view images portray a perspectively correct top-down view of the environment from directly above the robot.

(a) Omnidirectional view of DM-spqrSoccer2006_250.utx

(b) Omnidirectional view of DM-compWorldDay1_250.utx

(c) Perspective view of DM-spqrSoccer2006_250.utx

Figure 3: Images taken in the USARSim environment.



(a) transformation of figure 3(a) using Equations 1 to 3 with $z_w = 40$.

(b) transformation of figure 3(b) using Equations 1 to 3 with $z_w = 40$.

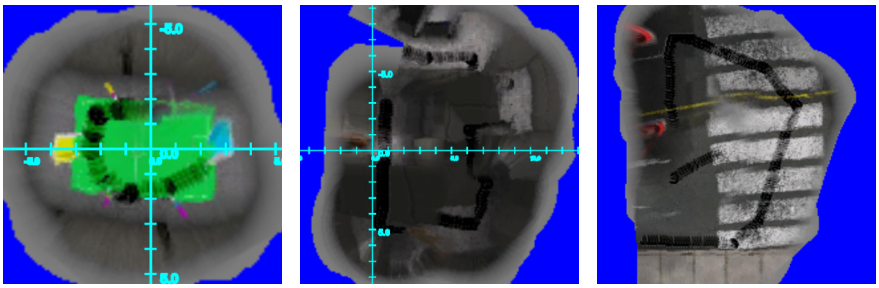
(c) transformation of figure 3(c) using Equations 4 to 6 with $c = 550$.

Figure 4: 500×500 pixel bird-eye transformations of Figures 3.

Ideally, all landmarks and soccer goals in Figures 4(a) and 4(c) would be depicted as if they were observed from far above as would be the case with a vertical orthographic projection of the environment. Unfortunately, orthographic projection cannot be performed on images produced by cameras which have a single effective viewpoint close to the ground. Perspective projection in combination with a relatively low position of the camera results in a depiction of landmarks and goals which is exceptionally stretched.

3.2 Mapping

The first map construction experiments where done without the obstacle filtering. They where performed in three different RoboCup worlds. Unfortunately, most of the RoboCup worlds resemble an environment after a disaster, which means that there color scheme is mainly gray. The constructed maps are not that sharp as the single images depicted in Fig. 4, because they are a combination of many bird-eye views, every time from a slightly different location. The constructed maps cover more area than a single bird-eye view. For instance Fig. 5(b) covers an area of 20x20 meter.

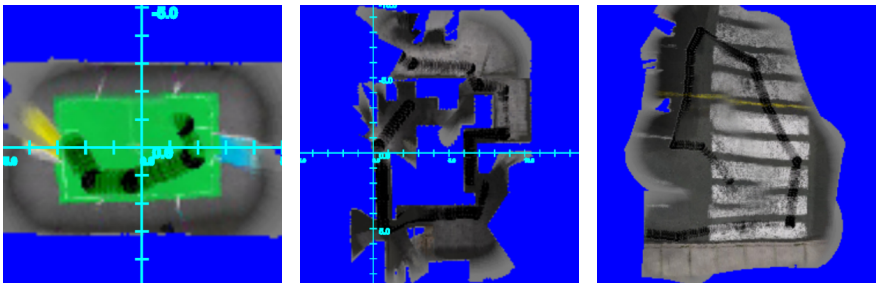


(a) DM-spqrSoccer2006_250 (b) DM-Mapping_250.utx (c) DM-compWorldDay1_250.utx

Figure 5: Bird-eye view maps created in 3 different RoboCup worlds, without object filter

The best resemblance to the real world was gained with the DM-spqrSoccer2006_250 map (Figure 5(a)), obviously because this is a relatively small map with hardly any obstacles. As can be seen in Figure 5(b), the obstacles (mainly the walls) oppose a huge problem in the final outcome of the map. They greatly overlap other parts of the map. This fact combined with the overall grayness of the world, results in a quite unclear and unusable map. For the map to be of any use, there must at least be a clear distinction between free space and non-free space. Since this distinction can not be made without any form of obstacle detection, the obstacle filter was introduced (see section 2.2.4).

When using the obstacle filter, one can clearly see a huge improvement in the created maps. Figure 6 shows the same maps as discussed above, only now the obstacle filter has been used. The greatest improvement can be found in the DM-Mapping_250 map (Figure 6(b)). The free space is now clearly visible, while the non-free space is not drawn at all, leaving the original blue color. This results in an image with clearly visible outlines of the actual world, while Figure 5(b) did not have this effect at all.



(a) DM-spqrSoccer2006_250 (b) DM-Mapping_250.utx (c) DM-compWorldDay1_250.utx

Figure 6: Bird-eye view maps created in 3 different RoboCup worlds, with object filter

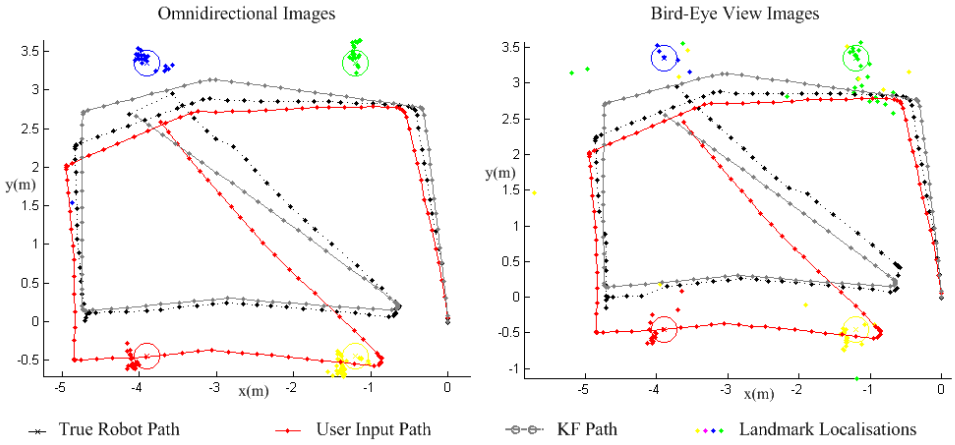
When taking a closer look at the generated maps using the objects filter, there are some details that need to be discussed. One typical error that seems to occur more often when the robot makes a lot of sharp turns, is the breaking of some straight lines. This can be seen quite clearly in the image from DM-compWorldDay1_250.utx. The lines of the zebra crossing are distorted and two divided parts of the map with a different rotation can be seen. This error can be attributed to 90 deg turn on the left hand side.

Another detail worth mentioning is the black trail that is visible on most parts of the map. This is the path the robot traveled. Because the center of the omnidirectional images will always exist of the robot, the robot will be drawn on every part of the map where an image is taken. If needed (depending on the application), this trail could be easily removed by including the robot shape as an obstacle in the obstacle

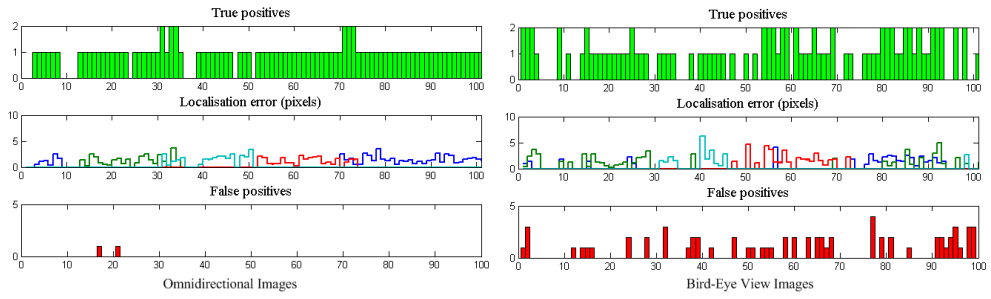
filter. The robot's image would then be filtered when applying the objects filter and it will no longer leave a black trail.

3.3 Self Localization

From the previous examples it is clear that bird-eye view maps can be created. What remains is the question how accurate these maps are. In previous work [11], self-localization experiments where performed to measure the accuracy. These experiments were based directly on the omnidirectional images. The same experiments are now performed, this time with bird-eye view images. The overall results from the experiments (Figure 7) show a lot of resemblance. In both cases the robot can estimate its location quite accurately. The location as estimated from the user input slowly drifts off, accummulating in an error up to a meter. Both Kalman filters reduce this error to a few decimeters. The performance on omnidirectional and bird-eye view images result are comparable. However, for the bird-eye view the stretching of the images as mentioned in section 2 results in larger landmarks. These larger landmarks are easier to detect, but small errors on the omnidirectional image will also be stretched and become larger. This results in more landmarks being detected simultaneously (see Figure 7(b)), which results in a better pose estimate and an overall better localization. Also the number of false positives increases; quite a few landmarks are found at wrong locations (sometimes meters off), compared to results with the omnidirectional images. Fortunately, these false positives do not influence the localization results, because such large detection errors can be removed with a validation gate [9].



(a) Path prediction results



(b) Landmark detection

Figure 7: Self Localization results in the DM-spqrSoccer2006_250 world for omnidirectional and bird-eye view images

4 Discussion and Further Work

Based on the theory described in section 2 and regarding the results in the previous section we can conclude that it is possible to create maps from a bird-eye view's perspective using an omnidirectional camera that are more intuitive and can provide an additional layer to the user interface of the operator of the robot. Such a layer with a bird-eye view map can be very valuable in environments where surfaces with a different level of traversability can be distinguished by their color or their texture. Furthermore, the bird-eye view images are suitable for self-localization algorithms and might even be more suitable than the default omnidirectional images. Since the landmarks are enlarged by the transformation they remain detectable over longer distances.

However, there is still enough room for improvement. Possibly the most important next step would be to fix the localization errors that occur, this would then result in maps more true to the actual world and thus more useful to rescuers. To fix these errors one could use the omnidirectional images for self localization in an unstructured environment. The image might even be used for object detection when accompanied by a learning algorithm which can learn to detect objects. This would then result in a robot that uses the omnidirectional camera for all basic navigation algorithms and uses the other sensors only as an extra aid for localization.

References

[1] S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):1–22, 1999.

[2] O. Booij, B. Terwijn, Z. Zivkovic, and B.J.A. Kröse. Navigation using an appearance based topological map. In *Proceedings of the International Conference on Robotics and Automation ICRA'07*, pages 3927–3932, Roma, Italy, 2007.

[3] R. Bunschoten and B.J.A. Kröse. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*, 19(2):351–357, 2003.

[4] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Bridging the Gap Between Simulation and Reality in Urban Search and Rescue. In G. Lakemeyer, E. Sklar, D.G. Sorrenti, and T. Takahashi, editors, *RoboCup 2006: Robot Soccer World Cup X*, volume 4434 of *Lecture Notes on Artificial Intelligence*, pages 1–12, Berlin Heidelberg New York, October 2007. Springer.

[5] K. Daniilides and N. Papanikolopoulos. Special issue on panoramic robots. *IEEE Robotics & Automation Magazine*, 11(4), December 2004.

[6] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000.

[7] H. Ishiguro and R. Benosman. Special issue on omnidirectional vision and its applications. *Machine Vision and Applications*, 14(2), June 2003.

[8] A. Jacoff, E. Messina, and J. Evans. A standard test course for urban search and rescue robots. In *Proceedings of the Performance Metrics for Intelligent Systems Workshop*, pages 253–259, 2000.

[9] Steen Kristensen and Patric Jensfelt. An experimental comparison of localisation methods, the mhl se ssions. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, pages 992–997, October 2003.

[10] Shree K. Nayar. Catadioptric Omnidirectional Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 482–488, Jun 1997.

[11] Tijn Schmits and Arnoud Visser. An Omnidirectional Camera Simulation for the USARSim World. In *Proceedings of the 12th RoboCup International Symposium*, July 2008. Proceedings CD. To be published in the Lecture Notes on Artificial Intelligence series.

[12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Inc., Cambridge, MA, USA, 2005.

Online Collaborative Multi-Agent Reinforcement Learning by Transfer of Abstract Trajectories

Maarten van Someren Martin Pool Sanne Korzec

Informatics Institute
Kruislaan 419
1098 VA Amsterdam
University of Amsterdam
email: maarten,skorzec@science.uva.nl

Abstract

In this paper we propose a method for multi-agent reinforcement learning by automatic discovery of abstract trajectories. Local details are abstracted from successful trajectories and the resulting generalized, abstract trajectories are exchanged between agents. Each agent learns a policy for its own environment. By abstracting trajectories and sharing the result the agents benefit from each others learning. This reduces the overall learning time compared to individual learning.

1 Introduction

Intelligent agents typically learn individually. In contrast, humans acquire their skills with a combination of practice and knowledge that they receive from others. Some of our experience with the world is passed on from one person to another in a student-teacher setting. Other knowledge is codified in books or in electronic form. As humans we typically learn skills by combining such knowledge with hands-on experience. One reason for this is that it is not possible to codify and retrieve all knowledge. We codify knowledge that is hard to acquire by experience and yet useful for solving a problem or acquiring a skill. This observation motivated the study of learning settings in which knowledge is both acquired by individual agents and shared between agents. We take a typical skill learning setting as context, Reinforcement Learning.

Standard Reinforcement Learning (RL) methods learn a policy for an environment by observing the effect of the actions and the rewards obtained from the environment. This policy is specific for the environment: the states, the effects of operators and the position and size of rewards. Here we consider learning when some parts or aspects are the same in all environments and others are different. Different agents are learning in these environments. They can benefit from each others learning by exchanging parts of policies that are useful in all environments. In abstract terms this problem is in many respects the same as that of a single learner in a single environment where policies that were learned in one part can be transferred to other parts but in this paper we focus on the multi-agent setting. Our goal is thus a method to speed up learning by transferring parts of policies, in a single environment and also between multiple agents in different environments. This paper describes the setting that is used and preliminary experiments.

Note that we do not consider multi-agent learning in the sense of performing a task collaboratively. Each individual agent has its individual task and environment and do not perform a task collaboratively. They only collaborate by sharing what they have learned: they *learn* together but do not act together.

In the literature we find several different approaches to the problem of transfer. The first is to apply generalization to (partially learned) policies. Instead of acquiring a policy by basic Q-learning, generalization is used to complete a policy. By learning the relation between (features of) states, (features of) actions and (features of) rewards, predictions can be made about policy entries for which little or no experience is

available. A second approach is to give the learner a predefined, possibly hierarchical, decomposition of the environment. The learner can then learn policies for parts of the environment and these can be used as compound actions to learn a total policy. The environment is decomposed into parts by specifying an initial state and a goal state for which the agent can learn a (sub)policy. The decomposition can be hierarchical. This underlies hierarchical Reinforcement Learning methods such as MAXQ [1]. This decomposition again will be useful if it captures components of policies that appear multiple times because this enables transfer. Once a new subpolicy is learned it can be used for learning new tasks, simply by treating it as a (macro-)action. The main difficulty here is to discover that certain parts of the environment are the same (ie. have the same policy), before their policies have been learned completely (see for example [7] for a discussion).

We consider a variation of this learning task in which there are several agents that live in different parts of a large environment. These parts are characterized by different geographical (or perhaps rather: topological) structures but governed by the same physical laws. This means that the environment cannot be divided into parts that match completely nor that a policy can be transferred 'as is'. Instead, abstract descriptions of states must be constructed that can be used in different environments for learning (sub)policies.

Our approach is based on the notion of *Abstract Trajectories*. An Abstract Trajectory is a description of a set of trajectories in terms of changes in a subset of the properties that define a state. An abstract trajectory corresponds to a part of an environment: the part that matches the property changes. Given an abstract trajectory and an environment, a policy can be learned for the part of the environment that matches the abstract trajectory. Thus, a learner can use an Abstract Trajectory by first learning a policy for states that satisfy the abstract trajectory and then incorporating the result in RL as an *option* [8], a kind of macro-operator. RL treats the option as an operator and includes it in its policy. This speeds up learning if the abstract trajectory is actually a useful macro–operator in the final policy. In the following sections we discuss how Abstract Trajectories are found (section 2) and how they are used in RL. Then we describe experiments to study the effect (sections 4 and 5).

2 Finding abstract trajectories

The environment for learning consists of a set of states. Each state is described by a set of properties. The agents know the possible actions for a state. Different agents live in different "sub-environments" that do not overlap. Abstract Trajectories are found by exploring the environment, here in the context of Q–learning [9]. An example of an abstract trajectory is:

$$key1 : no,\ key2 : no,\ key3 : no,\ door : closed \rightarrow$$

$$key1 : yes \rightarrow key2 : yes \rightarrow door : open$$

This describes a set of trajectories that is characterized by a change in the property *key* from value *no* to value *yes* and a change in property *door* from *closed* to *open*. In this example *key* opens the door.

An abstract trajectory is defined as a sequence of abstract states where an abstract state is a set of attribute value pairs. An abstract trajectory is used as a kind of learning goal. An agent learns a policy that enables it to follow the abstract trajectory in its own environment.

What is a useful trajectory and how is it abstracted? Our method is this. During learning, we store all trajectories. maintain the set of abstract trajectories. An abstract trajectory δ subsumes abstract trajectory ϵ if it can be constructed by removing states from δ. The algorithm to discover the abstract trajectories is given in Figure 1.

Infrequent Abstract Trajectories are discarded and very frequently occurring (using a threshold) Abstract Trajectories are sent to other agents. The rationale for this is that property changes that occur frequently in successful trajectories are likely to be useful for other agents in other environments. It would be possible to exchange Abstract Trajectories that consist of multiple property changes but it is less clear that these are useful for other environments. It is useful to first pick up a fire extinguisher and then extinguish the fire but in a different environment it the order in which an agent should pick up a key and a fire extinguisher may be different in different environments.

An agent that receives an Abstract Trajectory starts to learn a policy for the part of its environment that is covered by the Abstract Trajectory. The resulting option has states that satisfy the initial value of the property, before the change. This becomes the condition for applying the option. Once learning the policy for the option converges and the policy is stable, it is incorporated into RL. In this way the agent learns when to use the option.

Abstract trajectories

```
For each trajectory that ends in a goal state, perform the
following actions:

1. Remove properties that change frequently (using a threshold)

2. Remove all subsequences without property change, leaving only
the states between which a property changes.

3. Construct an abstract trajectory for each subsequence between
the initial state, the final state and each change of a single
property.
```

Figure 1: Discovery of Abstract Trajectories

Our Abstract Trajectories are a form of *state abstraction*, [3, 5, 6]. In our case, the goal is actually twofold: to find the best abstraction for the agent and also to find the abstraction that is useful for other agents that live in different environments. The current version simply generates an abstract trajectory per property. There are many other possibilities here that need to be explored in future work.

3 Collaboration and Communication

The organization of the agents is quite simple. An agent that has found a useful Abstract Trajectory broadcasts it to all other agents. The agents remember this and will not duplicate it. Communication is synchronous The system runs in cycles and in each cycle an agent moves to a new state, learns from this, checks if it found a good Abstract Trajectory and if so, broadcasts it.

There are many alternatives. For example, agents could exchange candidate Abstract Trajectories and see if *together* over all environments they are frequent enough. This gives more communication but this can be compensated by reduced learning time. In this study we choose to explore the simplest organization form.

4 Experiments

The purpose of the experiments is to evaluate the effect of learning and exchanging Abstract Trajectories. First, we perform a preliminary experiment to demonstrate the effect. Good *specific* trajectories are given to the agent and performance is compared with that of agents without them. Next we consider the effect of learning Abstract Trajectories. Finally we consider the effect of Abstract Trajectories in a multi-agent setting in which agents live in different environments. These environments vary in geography but they are subject to the same 'physics'. Therefore the optimal policies are different and exchanging specific trajectories or policies is not useful but some abstract trajectories are useful in all environments and therefore can contribute to the learning performance of other agents.

The agents perform regular undiscounted Q-learning, starting with $\epsilon = 1.0$ and multiplying by 0.99 (decreasing with 1%) at each episode.

4.1 Mazes

We have conducted our experiments with maze environments. An agent is in search of the exit, which can be blocked by obstacles. The agent needs to use items in the environment to get past the obstacles. For example, to pass a locked door, the agent will first need to pick up an item (e.g. a key) and use that item on the obstacle.

All are 8x11 mazes with a horizontal wall running through the middle. See for an example Figure 2. In the wall is an opening with an obstacle. The nature of the obstacles varies: two obstacles (river and

roadblock) just keep the agent from passing, one (monster) causes the agent to die if he doesn't have the proper object, and one (fire) does both: the agent may pass if he has the correct item (a fire extinguisher), but he will die if he also carries the associated 'evil item' (a jerrycan of petrol). The mazes also contain two irrelevant items. The mazes differ in the positioning of the starting point and exit. The exit is always in one of the four corners and the starting point is exactly on the opposite side of the maze. The size of the state space of the small mazes is 1408 states.

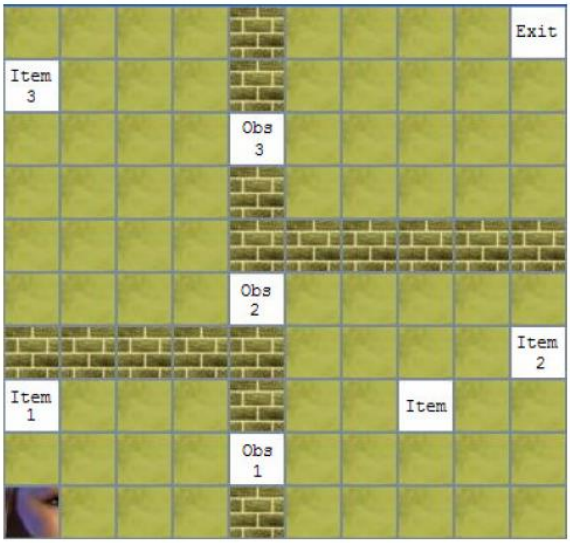


Figure 2: The large maze

The large maze (Figure 2) is 10x10. The path to the exit is blocked by a number of walls containing all but one of the obstacles. Also in the maze are all four of the solving items, making one of the items an irrelevant item. In addition, the large maze may or may not contain two more irrelevant items. The large maze has 12,800 states, or 51,200 states when irrelevant items are included.

4.2 Performance measure

We will use speed of convergence to optimality as our measure. More precisely, we say that convergence is reached at the first episode of the first 20 episodes of which the average reward is no more than 5 points away from the optimal reward.

5 Results

5.1 Experiment 1: The effect of Abstract Trajectories

The purpose of this experiment is to test if the Abstract Trajectories work at all. We use a simple 8x11 maze with one obstacle, one key item, and two irrelevant items. First an agent learns abstract trajectory and then acquires the policies for these trajectories. We manually add either the specific trajectories (with policies) or the Abstract Trajectories to a new agent and have it learn the environment. We expect these to learn faster than without abstract trajectories and the agents with specific trajectories should be faster than those with only abstract trajectories.

The comparison in speed of convergence is between:

Options Given: Options are given to the learner explicitly. This is a kind of reference.

Abstract Trajectories Given: In this version, the agent is given the correct Abstract Trajectories and learns its own policy for it.

Self-used Abstract Trajectory learning: The agent finds Abstract Trajectories and uses these in its own learning.

Q: Plain Q–Learning by a single agent without anything given, nor does the agent learn or use Abstract Trajectories

We ran each setting 10 times with a maximum of 600 episodes. The results can be found in Table 1.

Agent	Average
Options	142
Abstract Trajectories	177
Self-used Abstract Trajectory learning	208
Q	244

Table 1: Speed of Convergence

As can be seen from the table, the agent that is simply given the complete options performs best, converging to the best solution in 142 episodes on average. The agent that is given the Abstract Trajectories is next best. It has to learn the policies for the Abstract Trajectories which costs extra cycles. The agent that discovers and then uses its own Abstract Trajectories Needs a substantial number of extra cycles. Discovering the Abstract Trajectories is thus fairly expensive. A simple Q–Learner needs 244 cycles on average. This means that we may expect that the exchanging Abstract Trajectories can have a positive effect. In a setting where new agents enter the overall environment, an agent would do some 30% better if the agents in the environment would give it the right Abstract Trajectories.

5.2 Experiment 2: Learning Abstract Trajectories

In this experiment, we study the multi–agent setting where learning Abstract Trajectories takes place in parallel in all agents. We compare an agent that just does Q–Learning (*Q*) with an agent that is given useful Abstract Trajectories and only learns policies for them (*Abstract Trajectories*), with a agent that generates its own Abstract Trajectories and learns from this (*Self-used Abstract Trajectory*), and finally with an agent that learns along with four other agents. One agent, the target agent, is learning in a large maze (10 by 10) and four other source agents are in the four smaller mazes (8 by 11). In the smaller mazes Abstract Trajectories will be found earlier than in the larger maze. Distributing these to the other agents, including the one in the larger maze should speed up learning. All agents start learning at the same time. The four agents find the abstract trajectories by exploring their own environment and they transfer them to the agent in the large maze, which starts learning policies for them, in addition to his normal Q–learning, including learning policies for its own trajectories, *Multi–agent Transfer Learning*.

In addition, we vary the number of irrelevant items in the maze, and thus the number of irrelevant properties of states. We uses mazes with no, 1 and three irrelevant items. Figure 2) shows the maze without extra irrelevant items.

Agent	Average 1 irrelevant item	Average 3 irrel. items	
Multi–agent Transfer Learning	170		251
Self-used Abstract Trajectory	187	320	
Q	247	367	

Table 2: Convergence for the multi–agent transfer experiment.

Results for this experiment are in Table 2. The results confirm our hypotheses. A simple Q–Learner needs most cycles (some more than in the previous experiment because the maze is more complex). Finding and using its own Abstract Trajectories improves learning. Exchanging Abstract Trajectories between agents gives an extra speedup, despite the fact that the environments of all agents are different.

This is Especially when the state space gets larger and the main agent needs more episodes to learn his policy and his subgoals, the difference between transfer of Abstract Trajectories and generation by the agent himself becomes larger.

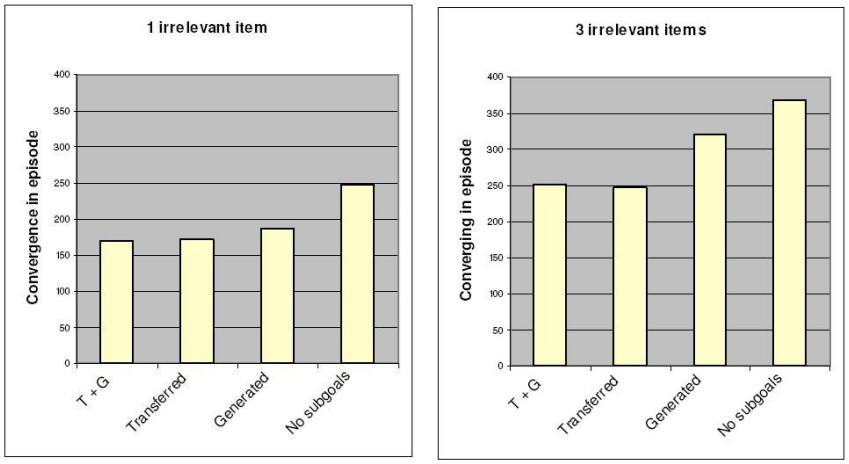


Figure 3: Results of the experiments. Left: Subgoal transfer with 1 irrelevant item. Right: Subgoal transfer with 3 irrelevant items.

The number of irrelevant items is likely to effect performance. The number of items determines the number of properties of the states and thereby the size of the state space. Because irrelevant items are less likely to appear in Abstract Trajectories that are transferred, the effect of discovering and transferring Abstract Trajectories is likely to be stronger in environments with more irrelevant items. This indeed what we see in Table 2. In particular the difference between the multi–agent setting and the single learner increases.

6 Related Work

Recent years have seen a strong increase in interest for methods that combine Reinforcement Learning and MDP estimation with other forms of Machine Learning. We position our work relative to earlier work that we find most similar.

Our method for constructing Abstract Trajectories is closely related to the HEXQ algorithm by Hengst [2] for discovering hierarchical representations for complex policies that use state abstraction and sequence aliasing. We use a hierarchy of only two levels and multi-agent instead of single agent learning. We do not integrate the values of states and let each learner find these by itself.

Barto and others [11, 6] describe a method for identifying and exploring commonalities between learned policies that is based on identifying types or classes of different objects that all appear in the same policies. A mapping is constructed that maps states and actions from one policy to another. It may involve mappings between objects that are part of states. Our abstract trajectories can be viewed as such a mapping. Finding such a mapping is a form of generalization because the mappings are used to construct a general class of objects or states. Our method is a weak form of this type of learning that searches for a rather simple mapping and leaves policy reconstruction to other agents.

Wilson et al. [10] study a transfer setting using a hierarchical Bayesian approach. MDPs are learned in the form of mixture models with an unspecified number of components. After learning models for some number of tasks or environments, the distribution of parameters of the models over these environments is estimated and used as prior for estimating the distributions of new environments. This approach exploits commonalities in probability distributions over classes as function of labelled states. In our setting this model is less appropriate because we assume a kind of dichotomy in the confidence of these distributions. Some parts of these distributions hold in all tasks and other hold in no other tasks.

Stone and his colleagues [7] study similar issues in the context of transfer learning. They focus on finding a mapping between tasks that enables a policy to be transferred between these tasks. For example, Jong and

Stone [3] search for generalisations over states that enable a simpler description of a policy preserving the agents behavior.

The main distinctive features of our method are: transfer between agents of goals, finding common states to guide abstraction, simple property-based abstraction and not transferring policies. Many of these methods are quite similar and our method can easily be extended to the existing schemes and vice versa. To decide which approach is best depends on the setting and in particular what different learning tasks have in common and what not.

7 Conclusion, Discussion and Further Work

The main contribution of this work is the demonstration of a basic and direct method for learning and transferring abstract trajectories between learning agents. Our method extends existing methods for Q-learning with options by discovering abstract trajectories that have no associated policy. The policy for an abstract trajectory must be learned. Abstract trajectories are generated automatically and used for further learning. Abstract trajectories do need additional learning of a policy but when they capture important aspects of the environment (in the form of Abstract Trajectories) then they focus learning on parts of the environment for which it is useful to create an option with a policy. This improves learning.

The current system is a prototype intended as a basic architecture and a proof of concept. It raises a number of new questions. Most of these have to do with the online character of the learning process. Our method uses a threshold for convergence to trigger exchanging abstract trajectories. There is no guarantee that the abstract trajectory at that point is optimal even for the sending agent, because the trajectory from which it was derived may not be optimal [4]. When a better path and abstract trajectory is found, the revision should be distributed to the other agents but they need to be able to revise the half-learned policy for the original abstract trajectory.

Constructing abstract trajectories can be viewed as a form of causal modelling and corresponding methods can be applied here. The relation between learning policies and learning causal models has been raised before in the context of a single learner. Our experiment suggest that abstracting from specific "subenvironments" to the world in general makes causal modelling attractive relative to learning policies. This suggests that the currently simple generalisation method can be replaced by a more sophisticated causal modelling algorithm. This setting will then be used to study coordinated learning by multiple agents. This involves questions of how the results of learning by different agents can be integrated. Another set of questions concerns the multi-agent coordination and communication: how can the agents optimally exploit each others partially learned abstract options? Who should send what to whom at what moment?

The current system uses a toy domain. Practical applications can be envisaged in the context where multiple Reinforcement Learners are used in a complex environment.

The experiments show that constructing shareable knowledge is feasible and effective in a standard Reinforcement Learning setting.

Acknowledgments

The authors want to thank Attila Houtkoper and Gideon Maillette de Buy Wenniger who developed the first version of the Multiquest system in 2004 and Michael Metternich who revised the original version in 2005. We also thank the reviewers for many useful comments that helped to put this work in perspective.

References

[1] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res. (JAIR)*, 13:227–303, 2000.

[2] Bernhard Hengst. Discovering hierarchy in reinforcement learning with hexq. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 243–250. Morgan Kaufmann, 2002.

[3] Nicholas K. Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 752–757, August 2005.

[4] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *CoRR*, cs.AI/9605103, 1996. informal publication.

[5] Andrew McCallum. Instance-based utile distinctions for reinforcement learning with hidden state. In *Proceedings ICML*, pages 387–395, 1995.

[6] B. Ravindran and A. G. Barto. Relativized options: Choosing the right transformation. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 03)*, pages 1011–1016, 2003.

[7] Peter Stone. Learning and multiagent reasoning for autonomous agents. In *The 20th International Joint Conference on Artificial Intelligence*, pages 13–30, January 2007.

[8] Richard S. Sutton, Doina Precup, and Satinder P. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell*, 112(1-2):181–211, 1999.

[9] Christopher J. C. H. Watkins and Peter Dayan. Technical note: Q learning. *Machine Learning*, 8(3):279–292, 1992.

[10] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings ICML*, pages 1015–1022, 2007.

[11] Alicia Peregrin Wolfe and Andrew G. Barto. Defining object types and options using mdp homomorphisms. In *Proceedings of AAAI-06*, 2006.

Imitation and Mirror Neurons: An Evolutionary Robotics Model

Eelke Spaak [a]
eelke.spaak@gmail.com

Pim F.G. Haselager [a]
w.haselager@nici.ru.nl

[a] *Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands*

Abstract

The involvement of the mirror neuron system (MNS) in both imitation and action understanding has been firmly established. Various authors have claimed that the MNS's function in facilitating imitation builds upon its role in action understanding and is a phylogenetically later development. We argue that this hypothesis lacks sufficient evidence and present support for the reverse: the phylogenetically primary function of the MNS is imitation and the MNS could have evolved in response to a selective pressure for imitative behavior. This hypothesis was tested using evolutionary robotics simulation techniques. The simulation was conducted with embodied and embedded agents with a lifetime-adapting neural network for which the learning parameters were evolutionarily optimized. The agents had to perform an imitation task. Analysis of the resulting controller revealed artificial neurons showing clear mirror characteristics, suggesting that, indeed, mirror neurons evolve due to a selective pressure for imitative behavior.

1 Introduction

Mirror neurons are a particular class of visuomotor neurons, originally discovered in the macaque monkey, that fire both when the monkey performs a particular action and when the monkey sees that same action being performed by another individual [2]. The mirror neuron system (MNS) has been described as the neural basis of action understanding, as the system responsible for the human capacity to imitate and as the crucial step in the evolutionary development that ultimately led to modern language (see [10]).

The evolutionary origin of the mirror neuron system itself remains, however, not well understood. Identifying the circumstances under which a system that seems so essential to our higher cognitive abilities has come to evolve will likely provide much insight into what makes us human.

In an attempt to characterize the MNS's evolution, Vilayanur Ramachandran [8] claims that the extensive role it plays in the modern human mind should be considered an exaptation (i.e., an adaptation whose current function is not the function for which it originally evolved). The absence of any direct evidence concerning the MNS's evolution does not warrant any dismissal of this claim (nor does it lend it any credibility), but Ramachandran's hypothesis does leave open the question of which selective pressure *originally* led to the MNS's evolution.

An alternative hypothesis that attempts to address this question directly has been put forward by Elhanan Borenstein & Eytan Ruppin [1]. Using a computational model of evolution, they artificially introduced a selective pressure for imitation learning. The evolutionary process resulted in agents being born with neurons that had 'mirror-like' properties. These findings suggest that, when a selective pressure for imitation learning is present, i.e., when the capacity to imitate is beneficial to an individual, mirror neurons tend to evolve to meet this pressure.

Borenstein & Ruppin's agents are, however, radically disembodied and their simulation was not conducted embedded within a realistically simulated environment. This drastically reduces the applicability of their claim to real-life biological organisms. It is essential of biological mirror neurons that they are intimately intertwined with both visual and motor processing and this important property was not taken into account in Borenstein & Ruppin's [1] study.

In this article, we investigate the evolutionary origins of the mirror neuron system. In particular, we attempt to address the question of whether mirror neurons tend to arise in a population left to evolve under an evolutionary pressure for imitation learning. First, we briefly review the literature concerning the function of the MNS. Next, we put forward a new computational model of the evolution of imitative behavior, in which artificial agents evolved to perform well on an imitation learning task. After this, we conclude by discussing the implications this new model has for the discussion concerning the MNS's evolution.

2 The mirror neuron system

Mirror neurons are visuomotor neurons, originally discovered in area F5 of the macaque monkey brain, that fire both when an individual executes a goal-directed action and when that same action is observed. Because of the invasiveness of the technique involved, there have not been extensive single-cell recording studies of the MNS in humans. There are, however, fairly conclusive indications from other experimental techniques, such as functional neuroimaging and electrophysiological recordings, that humans possess a mirror neuron system as well (see [10] for a recent review of the mirror neuron literature).

Rizzolatti, Fogassi, & Gallese [11] have reviewed and extended upon two main hypotheses concerning the function of the MNS that had been previously put forward in the literature: action understanding and imitation. The proposed MNS mechanism for action understanding is quite simple: when an individual observes an action, the mirror neurons representing that action fire, resulting in activation of the observer's motor system. This activation is highly similar to the activation that occurs when the observer is actually executing that action and, since the observer knows (one hopes) the consequences of his own actions, he will understand the action as it is being performed by the other. The proposed MNS mechanism for imitation is even simpler: when an action is observed, it results in a motor-like brain activation and motor memories are formed accordingly, allowing the observer to subsequently replicate the movement.

Empirical studies have shown that, indeed, the monkey and the human MNS are involved in both action understanding [13] and imitation [14].

2.1 Evolutionary origins of the MNS

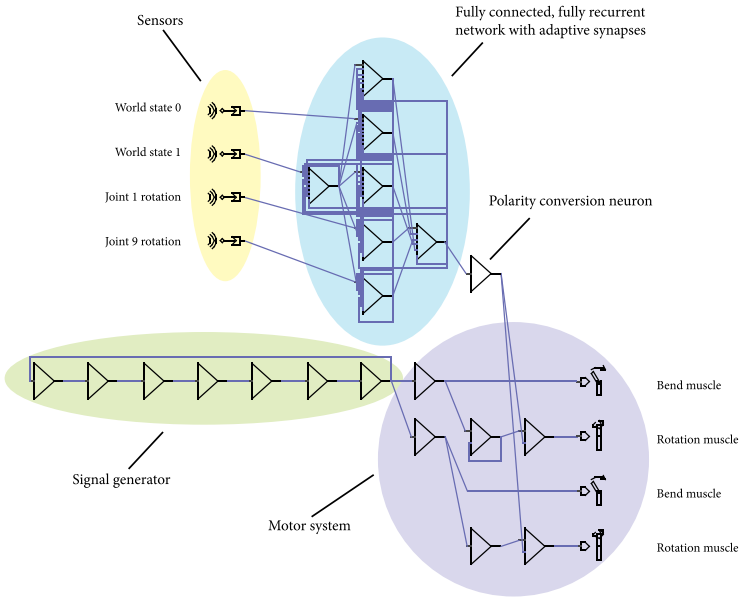
In order to understand the origins of any biological trait, one needs to consider the function it evolved to fulfil. This function can be identical to the function the trait fulfils today, in which case the trait is called an *adaptation*, or it can be different, in which case the trait with its current function is called an *exaptation*. The two functions the MNS most likely fulfils today, as described above, are action understanding and imitation, and Ramachandran [8] claims the most plausible evolutionary view is that the MNS has been exapted to fulfil these functions. However, in contrast to his view, in the absence of any clear evidence we believe our best bet is to try and identify a function the MNS is an adaptation for.

Rizzolatti has claimed that the MNS's role in action understanding phylogenetically predates its role in imitation [9]. However, it seems highly likely that imitation without understanding occurs in many animals, including humans. For instance, a flock of birds will often fly away in its entirety after one or two individuals have started to flap their wings [12] and newborn human and monkey babies are already able to imitate mouth gestures they observe [6]. In both cases, presuming a true understanding of the observed action seems questionable. Rizzolatti admits this, but his solution to the problem seems rather problematic.

He proposes to make a distinction between low-level and high-level resonance mechanisms. The low-level mechanism would have evolved much earlier than the high-level one, would be located "close to" the motor system and would be responsible for the kinds of imitative behavior mentioned above. The high-level mechanism (i.e., the F5 MNS) would have evolved much later and would be responsible for action understanding as well as imitation. Rizzolatti claims that this view provides a unitary account for the different types of imitative behavior, but in our view it introduces a distinction that need not be there. First, there is no clear evidence indicating the presence of a mirror-like low-level mechanism in addition to the F5 MNS. Second, while the MNS as a motor- and perceptually grounded basis for action understanding is an elegant view, endowing one type of mirror system with "cognitive meaning" (i.e., the capacity to understand) while depriving another of it negates much of this elegance – for whence could this extra meaning come from, if the basic mechanism is identical?

Since imitative behavior is present in phylogenetically 'lower' species, we believe it plausible the function the MNS originally evolved to fulfil is imitative behavior.

Figure 1: The artificial agents' neural controller.



3 The model

To test whether a selective pressure for imitative behavior can result in the evolution of mirror neurons, we constructed a model with evolutionary robotics simulation techniques [7]. The model we present was constructed within the Framsticks simulation platform [4]. Artificial agents, composed of a body and a neural controller, were subjected to evolutionary optimization. To describe the model, we first outline the structure of the body and controller of the agents, as well as the environment they inhabit. Following this, the specifics of the evolutionary algorithm are given, after which we describe the results of the evolutionary simulation and analyze the evolved agents.

3.1 The agent and its environment

3.1.1 Body

The agents used in the present model have a two-legged body that is able to move in two directions. Each of the agents' legs is driven by a bend and a rotation muscle. The movement of the agents was simulated by the Framsticks Mechastick physics engine [5].

3.1.2 Brain

In figure 1, the layout of the agents' neural controller is shown. It is composed of four more or less separate modules: a sensors module; a fully connected, fully recurrent adaptive network that is subject to evolution and in which lifetime learning can occur; a signal generator, responsible for generating a periodic signal that drives the walking motion of the agent; and a motor system, responsible for integrating the output of the adaptive network and the signal generator and actually moving the muscles accordingly. The nature of the sensors and what they sense is described along with the agents' environment in section 3.1.3.

The neurons of the signal generator and the motor system are all of the Framsticks built-in 'simple neuron' type [4] and the parameters governing their activation characteristics were not subject to evolution.

The seven neurons making up the adaptive network are of a completely different nature. Their incoming connections have weights that can be adapted according to Hebbian-like learning rules during the lifetime of an individual agent. This architecture is based on that proposed by Floreano & Urzelai [3] and extended upon by Borenstein & Ruppin [1]. The exact manner in which lifetime learning can and will occur is not fixed, but rather determined through parameters subject to evolutionary optimization. Also subject to evolutionary

optimization are the initial weights of these synapses, allowing for a mix of innate and acquired traits to express itself in the behavior of the agents. For a single neuron, each input synapse i is governed by four parameters: $w_{i,0}$ — the initial weight of the input (real value in the range $[0, 1]$); s_i — the connection sign of the input (-1 or 1); η_i — the learning rate for the synapse (real value in the range $[0, 1]$); and r_i — the learning rule for the synapse (integer value in the range $[0, 4]$). At time t, the neuron's output is computed as follows:

$$input_t = \sum_{i=0}^{n} w_{i,k} s_i o_{i,t-1} \tag{1}$$

$$output_t = \frac{1}{1 + e^{-input_t}} \tag{2}$$

Where $o_{i,t}$ represents the output of input neuron i at time t and the subscript k is a time-dependent index whose relation to time governs how often synapses are adapted.

In the present model, the relation used was $k = \frac{t}{20}$. This results in the synaptic weights being updated every 20 time steps. The updated weights are given by:

$$w_{i,k} = w_{i,k-1} + \eta_i \Delta w_{i,k} \tag{3}$$

The value of $\Delta w_{i,k}$ is determined by the learning rule used for the particular synapse whose weight is being updated. The synaptic parameter r_i governs which learning rule is applied:

$r_i = 0$ — No learning:

$$\Delta w_{i,k} = 0 \tag{4}$$

$r_i = 1$ — Standard Hebbian learning:

$$\Delta w_{i,k} = (1 - w_{i,k-1}) o_{pre} o_{post} \tag{5}$$

$r_i = 2$ — Postsynaptic Hebb rule:

$$\Delta w_{i,k} = w_{i,k-1}(-1 + o_{pre}) o_{post} + (1 - w_{i,k-1}) o_{pre} o_{post} \tag{6}$$

$r_i = 3$ — Presynaptic Hebb rule:

$$\Delta w_{i,k} = w_{i,k-1} o_{pre}(-1 + o_{post}) + (1 - w_{i,k-1}) o_{pre} o_{post} \tag{7}$$

$r_i = 4$ — Covariance rule:

$$\Delta w_{i,k} = \begin{cases} (1 - w_{i,k-1}) \mathscr{F}(o_{pre}, o_{post}), & \text{if } \mathscr{F}(o_{pre}, o_{post}) > 0; \\ w_{i,k-1} \mathscr{F}(o_{pre}, o_{post}), & \text{otherwise;} \end{cases} \tag{8}$$

where $\mathscr{F}(o_{pre}, o_{post}) = \tanh(4(1 - |o_{pre} - o_{post}|) - 2)$.

In these formulas, o_{pre} is the activation of the presynaptic neuron and o_{post} is the activation of the postsynaptic neuron, both values averaged over the time steps since the last weight update (i.e., averaged over the last 20 time steps).

This type of adaptive synapses has been "based on neurophysiological findings (...) [and] these rules capture some of the most common mechanisms of local synaptic adaptation found in the nervous system of mammalians [*sic*]" [3, p. 433].

For each adaptive synapse, each of its four properties are encoded onto the genotype and subject to evolutionary optimization. The adaptive network is fully connected and fully recurrent, i.e., each neuron receives incoming connections from each other neuron and from itself. In addition, four neurons receive sensory input, each from another sensor, and the output of one neuron is propagated to the motor system.

3.1.3 Task and environment

The task the adaptive agents have to master during their lifetime is the execution of correct actions for given states of the world they inhabit. The correct action for each state is not initially known to the adaptive agents; they can only infer it by observing the action that is being executed by another, non-evolving, agent and learning to link this action to the state the world is currently in.

More specifically, the world the adaptive agents inhabit can be in two different possible world states, world state 0 and world state 1. Each world state has a different action associated with it, but which action is associated with which world state is not fixed. The two actions are walking (1) in north-east direction or (2) in south-west direction. The correct action for the current world state is always being executed by a demonstrator creature, co-inhabiting the world with the adaptive agent.

The adaptive agent is equipped with two world state sensors with which it can sense the current state of the world. One sensor is responsive to world state 0, whereas the other responds to world state 1. When the actual world state is equal to the state the neuron is sensing for, the sensor's output will be 1.0. When this is not the case, the sensor's output will be 0.0.

In addition to the world state sensors, the adaptive agents are equipped with two joint rotation sensors, sensing the rotation of certain joints in the demonstrator creature. More specifically, the rotation in the X-dimension is sensed for joints 1 and 9, respectively. These specifics were chosen after analysis of the demonstrator's locomotion revealed that these variables were the only ones showing variation with time; in other words, all the other joints were not controlled by muscles and were completely stiff. X-rotation was found to always lie within the interval $[-\frac{\pi}{2}, \frac{3\pi}{2}]$, and to make this compatible with the neural network, requiring activations in the range $[0, 1]$, this value is scaled. The scaled output of the joint rotation sensors is given by:

$$output = \frac{rotation}{2\pi} + \frac{1}{4} \tag{9}$$

It should be noted that, through their sensors, the adaptive agents have no privileged access to any of the internal workings of the demonstrator; only to its external behavior.

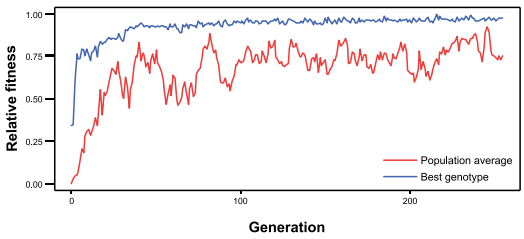
3.2 The evolutionary process

A population consisting of 20 adaptive agents, for which the properties of the adaptive network's synapses were randomly initialized, was subjected to evolutionary optimization. To prevent a correct world-state-to-action mapping from being genetically 'learned', each genotype was evaluated during two lifetimes, one for each possible mapping. At the beginning of each lifetime, the weights were initialized according to the initial weight values encoded in the genotype, so no properties acquired during the first lifetime could be expressed in the second. Each lifetime took $60\,000$ time steps. The world state was changed every $2\,000$ time steps. The world state was visible to the agent in 70% of its lifetime and invisible in 30% of its lifetime. The demonstrator executing the proper action was visible to the agent in 67% of its lifetime and invisible in 33% of its lifetime. Furthermore, the world state and the demonstrator's joint rotation were always visible in the first $12\,000$ time steps of the agent's life, simulating an infancy phase. These values ensure a good mix of different conditions is experienced during each lifetime.

During the evolutionary simulation of a single generation, each genotype is assigned an error score, rather than a fitness value. Fitness values are, for technical reasons, linearly computed from these error scores only at the end of each generation. Except during infancy, an agent's error score is updated just before each world state change. The angle (in the range $[0, 2\pi)$) of the path traveled by the agent is computed, as well as the angle of the path traveled by the demonstrator creature. The squared difference between these values is added to the error score. When the world state is invisible, the error score is updated by instead adding half of the absolute distance traveled by the agent to it. Summarizing, the fitness function rewards agents that perform the correct action for a visible world state and do nothing at all when the world state is invisible.

At the end of a generation, the next generation is created by making two exact replicas of the best genotype and selecting eighteen parent genotypes for reproduction according to a roulette wheel selection mechanism. Mutation occurs with a 60% chance; a mutated genotype will have a single property value changed to a random value within the valid range for that property type. The relatively high mutation chance of 60% ensures that enough genetic variation is introduced into the population, although offspring genotypes will still resemble their parent genotypes to a large degree, because each genotype has a large number of properties that can be mutated and, at most, only one of them is mutated during replication.

Figure 2: Fitness development over time.



3.3 Results

3.3.1 Fitness

The evolutionary simulation was run until 256 generations had been evaluated. The development of the population's average and maximum fitness value is shown in figure 2. A linear regression analysis was performed to estimate the effect of generation on average fitness. This effect was found to be significant ($F(1, 254) = 165.201, p < .001$) and strong ($r^2 = .394$). Average fitness increases with generation (standardized $\beta = .628$). Also, a linear regression analysis was performed to estimate the effect of generation on maximum fitness. This effect was found to be significant ($F(1, 254) = 152.563, p < .001$) and strong ($r^2 = .375$) as well. Maximum fitness increases with generation (standardized $\beta = .613$).

These results indicate that the evolutionary simulation was successful; agents capable of imitative behavior have evolved.

3.3.2 Neurodynamics

In order to determine whether or not neurons with mirror-like properties have evolved, the dynamics of the adaptive neural controller of the best individual of the last generation were analyzed. Interpreted in terms of a neural network working with activation values rather than action potentials, a 'mirror neuron' should be defined as a neuron showing the same activation pattern during execution of a particular action as during observation of that action.

During an agent's lifetime, it has to execute without observation when the world state is visible, but the demonstrator is invisible. Observation without execution occurs when the world state is invisible and the demonstrator is visible. Activation patterns of the seven neurons in the adaptive network, in both these conditions, for both world states, are shown in figure 3.

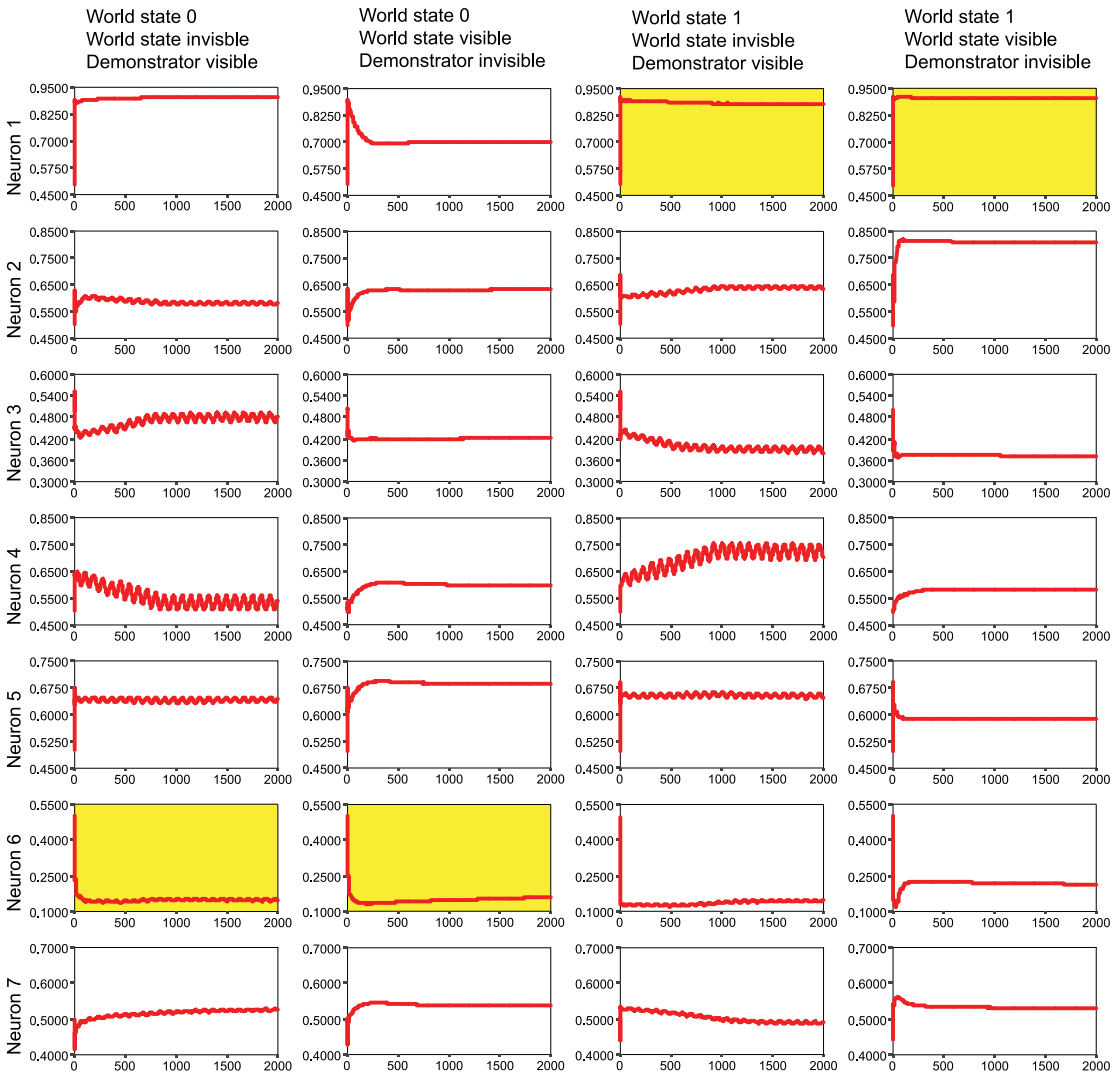
To quantify the mirror neuron definition in terms of statistical analyses of activation patterns, a mirror neuron is defined as a neuron showing a very strong correlation ($r > 0.85$) between its activation pattern in the 'observation, no execution' condition and its activation pattern in the 'execution, no observation' condition. Furthermore, the regression line relating the two activation patterns is required to have an intercept close to zero ($B_{constant} < 0.05$). Calculating these measures for all the neurons in both world states revealed that neuron 6 satisfies the mirror neuron criteria in world state 0 ($r = 0.910, B_{constant} = 0.033$) and that neuron 1 satisfies the mirror neuron criteria in world state 1 ($r = 0.885, B_{constant} = 0.030$). No other neurons were found to satisfy the criteria.

It should be noted that neurons with a fixed activation pattern, i.e., unresponsive to input variation due to very low incoming connection weights, could evolve in an evolutionary simulation. These types of neurons would also satisfy the criteria specified above, but the correlation between the activation patterns for different world states would also exceed the threshold. The neurons we found that satisfied the criteria did not show an above-threshold correlation between activation patterns for different world states, so we believe the conclusion that the findings are not due to chance is justified. A more formal error rate analysis of the procedure described above would be welcome, but is beyond the scope of this paper.

4 Conclusion and discussion

In the present model, artificial neurons emerged that have mirror-like properties. This shows that a selective pressure for imitative behavior *can* be sufficient for the evolution of mirror neurons. The primary function

Figure 3: The activation levels of the seven neurons in the adaptive network, plotted for 2 000 subsequent time steps in which the environment parameters were the same. In each graph, the X-axis represents time and the Y-axis represents output. Notice that a different range of Y values is plotted for the different neurons. This was done to improve clarity. However, to avoid a distorted view of the results, the Y range was kept constant for each neuron between conditions. Highlighted are the neuron activations for which statistical analysis revealed they satisfy mirror neuron criteria (see main text for details).



of the MNS could thus be imitation, rather than action understanding, and this latter function could be a phylogenetically later development that builds upon a capacity to imitate.

Once the function the MNS fulfils in facilitating imitation is considered as phylogenetically primary, one can avoid the highly problematic distinction between low-level and high-level resonance mechanisms that Rizzolatti [9] has proposed in order to explain imitative behavior in 'lower' species.

While the present model is a lot more realistic than previous models (such as that put forward in [1]), there are still some important factors that can be improved. Most notably, the artificial neurons used in the model employ continuous activation levels, whereas biological neurons use the frequency of their action potentials to transfer information. Because of this, the original definition of a mirror neuron is not applicable to the artificial neurons used in the model: the artificial neurons do not have a firing rate, while biological mirror neurons are defined by their relative firing rate in different conditions.

Finally, it should be mentioned that the results of the present study obviously do not prove that at some time in our evolutionary history, a selective pressure for imitative behavior occurred and that, in response to this, the mirror neuron system evolved. The present study *does* show, however, that this is quite a viable hypothesis and, we believe, one that is worth pursuing further.

References

[1] E. Borenstein and E. Ruppin. The evolution of imitation and mirror neurons in adaptive agents. *Cognitive Systems Research*, 6:229–242, 2005.

[2] G. DiPellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91(1):176–180, 1992.

[3] D. Floreano and J. Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13:431–443, 2000.

[4] M. Komosinski. Framsticks: A platform for modeling, simulating, and evolving 3D creatures. In A. Adamatzky and M. Komosinski, editors, *Artificial Life Models in Software*, pages 37–66. Springer, London, 2005.

[5] M. Komosinski and S. Ulatowski. Framsticks manual. Retrieved December 20, 2007, from `http://www.frams.alife.pl/common/Framsticks_Manual.pdf`, 2006.

[6] A. Meltzoff and M. Moore. Imitation of facial and manual gestures by human neonates. *Science*, 205:217–219, 1979.

[7] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, Massachusets, 2000.

[8] V. S. Ramachandran. Mirror neurons and imitation learning as the driving force behind "the great leap forward" in human evolution. *Edge*, 69, 2000.

[9] G. Rizzolatti. The mirror neuron system and imitation. In S Hurley and N Chater, editors, *Perspectives on imitation: From neuroscience to social science*, pages 55–76. MIT Press, Cambridge, Massachusets, 2005.

[10] G. Rizzolatti and L. Craighero. The mirror neuron system. *Annual Review of Neuroscience*, 27(1):169–192, 2004.

[11] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9):661–670, 2001.

[12] W. H. Thorpe. *Learning and Instinct in Animals*. Methuen, London, 1963.

[13] M. A. Umiltà, E. Kohler, V. Gallese, L. Fogassi, L. Fadiga, C. Keysers, and G. Rizzolatti. I know what you are doing: A neurophysiological study. *Neuron*, 31(1):155–165, 2001.

[14] A. Wohlschläger and H. Bekkering. Is human imitation based on a mirror-neurone system? some behavioural evidence. *Experimental Brain Research*, 143(3):335–341, 2002.

The Virtual Storyteller: Story Generation by Simulation

Ivo Swartjes Mariët Theune

University of Twente, P.O.Box 217, 7500 AE Enschede

Abstract

The Virtual Storyteller is a multi-agent framework that generates stories based on a concept called emergent narrative. In this paper, we describe the motivation and approach of the Virtual Storyteller, and give an overview of the computational processes involved in the story generation process. We also discuss some of the challenges posed by our chosen approach.

1 Introduction

The Virtual Storyteller is a multi-agent framework for generating stories [7].[1] These stories emerge from a simulation of virtual characters in a story world. Story generation happens in two phases: (1) simulation and (2) presentation (see figure 1). In the simulation phase, Character Agents work together with a Plot Agent to produce an interesting event sequence. The Character Agents simulate a character's life in the story world. They pursue goals, reason about their perceptions, experience emotions and make decisions in the context of this world. In order to constrain the authoring of engaging characters to a manageable degree, we focus on a particular domain of pirate stories. The Plot Agent facilitates the simulation by starting up scenes that specify the initial state of characters and story world for the particular scene. The event sequence resulting from the interaction of the Character Agents is captured by the Plot Agent in a formal representation (the fabula) that forms the input for the presentation phase. The World Agent manages a knowledge representation of the virtual world, executing actions and events as they occur, and sending back the results to the Plot Agent.

In the presentation phase, the Narrator component turns the formal fabula representation into an actual story by selecting the content to tell from the fabula, and applying language generation techniques to it in order to produce a text [8]. Optionally, the Narrator can be coupled to a talking face that tells the story using speech synthesis, thus simulating a human storyteller [2].

The rest of the paper is organized as follows. Section 2 gives a motivation of our work. Section 3 describes the theoretical foundation of our story generation approach. Sections 4 and 5 then describe the technical aspects of the simulation and presentation phase of the Virtual Storyteller, respectively. Finally, section 6 provides some conclusions and future work.

2 Motivation

With the advent of virtual reality and progressing computer game development, the computer as a medium is opened up more and more to the possibility of new forms of storytelling. Different from traditional media, the computer affords *interactivity*, adding a new dimension to the experience of stories. The interactor can potentially play an active role in storytelling by becoming a *participant*, determining the course of the story she takes part in. That this is a promising venture, can be seen by the fact that the computer games industry attempts to reconcile story with game play on a regular basis. However, how to do this well from a design perspective remains a difficult question. The reason for this is that there is a problematic tension between interactivity and story. This tension is sometimes called the *narrative paradox*: offering the interactor choices

[1] The Virtual Storyteller is implemented in Java on top of the JADE agent framework, using SWI-Prolog for its planning and reasoning processes.

breaks right through the notion of carefully orchestrated plot [1]. Authoring a customized continuation of the plot for all possible interactions quickly leads to an 'explosion of endings' and is untenable when the number of interactions increases.

Similar issues arise when interactive storytelling is used for educational purposes, such as scenario-based training. Here, educational goals are translated into a narrative environment that provides the interactor the chance to practice with problematic situations and explore the consequences of possible choices to make. Again, the system has to offer many different outcomes based on the choices of the interactors.

The narrative paradox is one of the fundamental issues that the Interactive Storytelling field is trying to address. One particular approach is offered through the concept of *emergent narrative* [1]. Emergent narrative is a simulation-based approach characterized by the fact that there is no predetermined plot. Instead, stories emerge through the interactions of virtual characters. There is also no longer one story; each character experiences his or her own story.

In the Virtual Storyteller project, we investigate under which conditions stories emerge through character interactions, with the characters being played by intelligent agents. An assumption we make is that we can later replace one of the characters in this setup by creating an interface for a human interactor without having to change much in the rest of the architecture.

3 Theory

For the generation of stories, our starting point is to look at how people comprehend stories. The reason to do this is threefold. First, story comprehension theory poses implications for system design because it offers narrative structures to aim for, informing amongst others the design of the cognitive processes of the characters. Second, the Virtual Storyteller can model its own understanding of the emerging story in order to make informed decisions on how it can develop further in the simulation phase. Third, the Virtual Storyteller can make assumptions about how the reader will comprehend the stories it produces, which informs decisions in the presentation phase.

There has been considerable research carried out on the subject of story understanding. Particularly influential research is described in [9], presenting a model that can be used to classify the clauses of a story into a small set of elements, and describe their causal connections. The model subsumes several other story understanding theories (e.g., story grammar and script theories). We use the theory to model the fabula of the simulation (i.e., what happened in the simulation and why). Our fabula model defines causal relationships between seven types of elements: setting elements, goals, actions, outcomes, events, perceptions and internal elements (see figure 2). A *setting element* describes an aspect of the story world that all the characters share and consider to be true. A *goal* is the main drive for a character to act. A goal in this context describes a desire to attain, maintain, leave or avoid certain states, activities or objects [9]. The state of the story world is changed through *actions* and *events*. The difference is that actions are performed intentionally by a character, whilst events are not; they just happen. An *outcome* is a mental concept that relates a goal to its fulfillment or failure. When a character believes that one of its goals is fulfilled, the goal has a positive outcome, but if the character believes that the performed actions did not succeed in fulfilling the goal, the

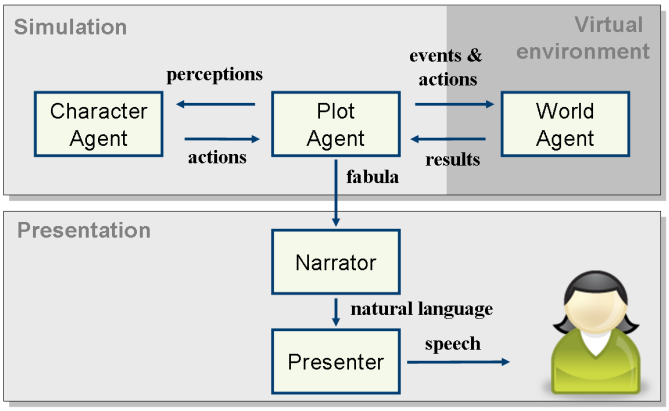


Figure 1: Global architecture of The Virtual Storyteller

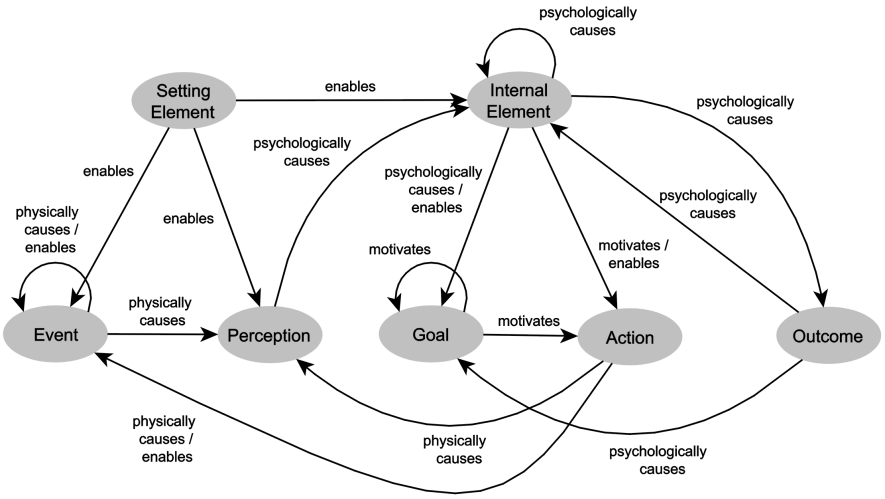


Figure 2: Fabula Model

outcome is negative. A *perception* describes properties of the world that a character witnesses. Everything else that goes on within a character, such as cognitions, emotions, feelings and beliefs, is classified as an *internal element*.

In a fabula, these elements are connected by four types of causalities: *physical and psychological causality* represent unintentional causes of either physical or mental nature, *motivation* represents an intentional cause and *enablement* represents any cause where the causing element is making it possible for the resulting element to occur.

For illustration purposes, figure 3 shows a small example story produced by the Virtual Storyteller using a simple simulation with one character.[2] The following sections explain how stories like this one are produced. Underlying this particular example story are actions such as opening the hatch, sailing to an island, and filling a water supply, motivated by a goal to refill the water supply and enabled by setting elements and beliefs about the existence and location of the island, and a pond on the island. The goal to refill the water supply is psychologically caused by a perception that the water supply is empty, which is physically caused by the event of the water supply running out of water.

Once upon a time there was a pirate, who was called Billy Bones. He was in the hold of his ship. The water supply was empty and he wanted to fill it. Therefore he opened the hatch. With a ladder the pirate walked to the deck. With the ship he sailed to an island. After he had gone ashore at the island, he filled the water supply with water from a pond.

Figure 3: Example story produced by The Virtual Storyteller (translation)

4 Simulation

At the basis of story production lies the simulation of the story world. The aim of the simulation is to produce a fabula that contains interesting event sequences for the presentation phase. The fabula production is done by modeling virtual characters whose interactions with each other and the story world determine how the story develops. The simulation is based on a multi-agent system in which Character Agents play out the role of the characters they represent, and share responsibility over plot progression. The design of these Character Agents is informed by improvisational theater [6].

The simulation is divided into rounds. Each round, the Plot Agent takes the initiative and requests each Character Agent to select an action. In response, each Character Agent goes through a deliberation

[2]The Narrator produces Dutch text. The original text is: *Er was eens een piraat, die Billy Bones heette. Hij was in het ruim van zijn schip. De watervoorraad was op en hij wilde hem vullen. Daarom opende hij het luik. Met een ladder liep de piraat naar het dek. Met het schip voer hij naar een eiland. Nadat hij bij het eiland aan land was gegaan, vulde hij de watervoorraad met water uit een vijver.*

cycle and responds with an action it wants to perform, or refuses the request, for instance because it is already performing an action, or because it cannot think of any action. When an action finishes, its effects are translated into perceptions that each Character Agent receives. The Character Agent is also informed whether an action was successful or failed.

4.1 Late commitment

One of the difficulties with the use of simulation for the emergence of stories is that the emerging stories are very much dependent on the initial setup of the story world, i.e., which characters, objects and relationships are defined. It is difficult for an author to try to imagine the effects of the way she sets up the story world in terms of the stories that come out and yet, informed decisions must be made. For instance, should there be a crate of rum bottles in the hold? Should two pirates hate each other? Or distrust each other? Is the ship they are on already carrying lots of gold, or is it empty? These kind of details might come to play an important role in a particular story, or they might not.

What makes the Virtual Storyteller unique in this respect, is that it is designed so that these details can be filled in as they become useful for the story under development, so that the initial setup can be kept small and basic. Based on the observation that actors in improvisational theater also do not work with a predermined, agreed upon story world, but frame this world as they go along [5], we have developed a technique that enables the use of this kind of decisions using operators that we call *framing operators*. Framing operators are a special kind of operators; they add new knowledge to the state of the story world (not only the physical state such as the objects and locations, but also the social state such as properties and relationships of characters) whilst creating the illusion that the knowledge has already always been present. In effect, framing operators implement a way for the character agent to say for instance: "Let's pretend I was the captain of the ship," or "Let's pretend I couldn't swim". This takes away the burden of having to determine the complete state of the story world in advance, and allows the agents to fill in particular details when useful. For instance, if it comes to be that the pirate ship is attacked by another ship, it is interesting if it is loaded with gold and the Virtual Storyteller can introduce this fact at the appropriate time. The decision to use framing operators is currently deeply interwoven with the action planning and goal selection processes of the characters. In the fabula, the effects of an executed framing operator end up as setting elements.

For instance, the starting point for the example story of figure 3 was Billy Bones' use of late commitment to justify the adoption of the goal to refill the water supply. This goal has preconditions stating that the character adopting the goal is the ship's captain (since captains carry this responsibility), that the ship has a water supply, and that the water supply is empty. Planning to establish this context, the Character Agent has used two framing operators (`BeCaptain` and `ShipHasWaterSupply`) and an event (`OutOfWater`). Figure 4 shows the `BeCaptain` framing operator in more detail.

We have also addressed some of the implications the use of late commitment has for the consistency of the simulation. All characters should accept the effects of a framing operator and be able to pretend they have always been true, so the characters should not treat the information as if it were new by for instance responding emotionally. To this end, we have implemented a mechanism to check if all characters can consent to the execution of a framing operator whenever selected by a character.

BeCaptain	
Preconditions:	*Effects:*
`ship(?ship)`	`captain(?me)`
`character(?me)`	`owns(?me, ?ship)`
¬ `captain(?me)`	
¬ `( captain(?other)`	
∧ `owns(?other, ?ship) )`	

Figure 4: The `BeCaptain` framing operator. Whenever there is a ship, and a character that is not a captain (of any ship) yet, and the ship does not have a captain already, the `BeCaptain` framing operator can define that the character is the captain of the ship and owns it.

4.2 Domain modelling

The domain of a particular story world is made up by representing the *setting* of the story world (the initial state of the simulation), and providing *schemas* that contain process-based information used by the cognitive processes of the characters.

The setting is based on an ontology describing the particular concepts of the story world. For our pirates world, this ontology contains concepts like `Sword` and `Pirate`. It also defines relationships relevant to the domain, such as `hates` and `owns`. In the setting, we use these concepts and relationships to represent the knowledge about the story world as it is in the beginning of a scene.

The schemas are data structures representing the actions that characters can perform, but also external events, character goals, scene changes and framing operators. Actions, events and framing operators are STRIPS-like operators that have preconditions specifying when the operator is possible, and effects that specify how the world changes as a result of executing the operator. For instance, an action schema `SailToLand` might have the effect that the character's ship is now moored at a nearby island, and as preconditions that the character is the captain of the ship, and that the ship is not already moored somewhere. Goal schemas model what the character wants and when. Goal schemas have preconditions that determine when an agent would adopt the goal described by the schema, success-conditions that determine when the goal was achieved and failure-conditions that determine when the goal was unsuccessful. Belief schemas model some of the reasoning processes a character might go through. They are executed whenever their preconditions hold, and thus act as simple rules. For instance, a belief schema might specify that if the hatch to the hold is open, and the character has not opened it himself, then a character can adopt the belief that someone is in the hold.

An important point in the modelling of the domain is that the preconditions form the context for behaviour. This context gains coherence by not only considering the state of the story world, but also relating to past events. For instance, a pirate's goal to raise the sails can have as precondition that the captain has just ordered him to do it. The Character Agents currently use the fabula knowledge representation as a model of episodic memory to store knowledge about their life in the story world.

Modelling the domain of a particular story world is a difficult task with no clear methodology as of yet. Developing a particular story world raises questions about which actions, events and goals to add, and how to represent their preconditions and effects in terms of a knowledge representation of the story world state. The simulation is *dramatic*, meaning that we make simplifications and abstractions from the perspective of drama, rather than that of a phenomenon under study. This makes our design effort very pragmatic; we make no claims on cognitive plausibility or validity of the simulated world and its characters. A story world is a dramatic abstraction of the real world and one must make choices in level of detail, and make assessments about possible interactions, so that one can cover enough of the story world in terms of agent behaviour.

4.3 Character Agents

As characters in a story, the agents of the Virtual Storyteller go through a deliberation cycle in which perceptions of actions and events lead to beliefs (a type of internal element), which in turn produce emotions (another type of internal element). These emotions then lead to behaviour in terms of actions and goals for which actions can be planned.

At the core of this deliberation cycle is a partial order planner that can determine which actions to perform to reach a particular goal. It fulfils one of the basic requirements for believable characters, namely goal-directed behaviour [3]. The algorithm is similar to standard partial order planning [4, ch.11], with the following modification: if the planner cannot fulfil an open precondition for the plan using the actions it knows, it can select either an event or a framing operator. Events and framing operators are "out-of-character" actions. They are not performed by the character in the story world, so it might seem strange to include them in the character's planning process. After all, a pirate character making a plan to become rich by going to a nearby cave just so he can "accidentally" discover a hidden treasure, does not seem to be very believable. This problem occurs if the Character Agent were to execute actions that are only in the plan in order to fulfil an open precondition of a framing operator or event. Therefore, we disallow plans in which such actions occur. If an action fulfils open preconditions of a framing operator or event, but was not added to the plan for this reason (e.g., the pirate went to the cave to seek shelter from the rain), the plan is allowed.

Another important issue in the deliberation cycle is goal management. Goals can be adopted whenever their preconditions are fulfilled. At any point in the simulation, the Character Agent might have many adopted goals, and is committed to the achievement of one of them. Goals, and their outcomes, anchor

episodic structures commonly found in stories [9]. For instance, successive episodes occur when the outcome of one goal leads to the adoption of a new goal. Such episodic structures, if aimed for in the generation of stories, have implications for goal management. We use the criterium that adopted goals, beside having their preconditions met, should at least also be causally related to whatever goal(s) the agent has already adopted or acted upon. We can check this by considering the preconditions of new goals to adopt; if these preconditions are fulfilled by fabula elements that are causally related to earlier goals, we have a causal connection between this new goal and the enabling or causing fabula elements.

The Character Agents play an important role in fabula production. An implication of the aim to create fabula is that the agents must be able to explain the causality of their decisions. They need to be able to explain for instance which goal motivated their actions, which beliefs enabled them, and which perceptions caused these beliefs. We do this by propagating this responsibility over the various processes that the agent runs. For instance, the partial-order planner can explain which goal motivated the planned actions, and which beliefs fulfilled the preconditions of the action selected for execution. The system gives each element (belief, goal, action) a unique identifier, so that once the element is established in the fabula, it can also be used later for subsequent causal explanations.

4.4 Plot Agent

The final responsibility for the simulation of the story world is with the Plot Agent. The Plot Agent has a facilitating role in the simulation, both outside and inside the simulated world. It exerts global control over the simulation by synchronizing its rounds. It captures the fabula elements that the Character Agents produce during the simulation and organizes them in a causal network for subsequent narration. It is also able to scale up the simulation by using *scene definitions*. A scene definition determines the setting, characters and their goals for a part of the simulation. The simplest story world simulation would consist of one scene defining the start state of the story world, and defining which characters occur in that story world. To start up a scene, the Plot Agent starts up new Character Agents if necessary, and casts them to play the required roles.

5 Presentation

Whereas many other systems in the field of Interactive Storytelling use 3D graphical environments to visualize the characters and the story world, in the Virtual Storyteller natural language is used as a medium to present the story to the user. The raw content for a story emerges from the simulation, as described above, in the form of a fabula. The Plot Agent passes this fabula on to the Narrator, which uses it to generate a story text. Optionally, the story can be narrated by an embodied agent using text-to-speech.

5.1 From fabula to document plan

The first step carried out by the Narrator is to determine the content and the global structure of the text to be generated. This is done by converting the input fabula to a 'document plan': a binary branching tree containing selected elements from the fabula, connected by rhetorical relations (relations between the parts of a text). Constructing the document plan involves removing those fabula elements that will not be explicitly expressed in the story. These include positive outcomes and beliefs caused by perceptions, which are considered to be inferable by the reader. It also involves adding new elements that represent background information about the story world or properties of the characters, such as their names and locations.

When mapping the fabula to a document plan, the causal links connecting the selected fabula elements are replaced with corresponding rhetorical relations. Consecutive actions motivated by the same goal are connected using a Temporal relation, which can be signalled using cue phrases such as *then* and *after*. Motivation and psychological cause links are mapped to Volitional Cause, which implies a certain extent of intentionality on the character's part. Cue phrases signalling this relation include *because* and *therefore*. Enablement and physical cause links are mapped to Non-volitional Cause, signalled by cue phrases such as *so that* and *thereby*. To introduce characters from the setting elements, the storytelling-specific Temporal-once relation is used, cued by *Once upon a time*. The Elaboration relation is used for background information, which is often expressed in a relative clause (e.g., *who was called Billy Bones*). The most general rhetorical relation is Additive (cue phrase: *and*); it is used if two fabula elements together cause another fabula element, and in general to connect two fabula elements if no other relation applies to them.

Next, the fabula elements in the document plan are replaced with abstract sentence structures called Dependency Trees. For each type of fabula element, a template is available specifying how its arguments should appear in the corresponding Dependency Tree. For example, the agens and patiens of an action are normally given the grammatical roles of subject and object, while instruments are expressed by a prepositional argument. An example from figure 3 is *He filled the water supply with the water from a pond*.To express internal elements, there are templates for standard sentences such as *The pirate was angry* but also for storytelling-style constructions such as *He had never been so angry!*, to be used for emotions with a high intensity.

To achieve coherent output texts that are more than a sequence of simple sentences, the Narrator may combine some Dependency Trees to form complex sentences. Whether it is possible to combine two Dependency Trees depends on the cue phrase selected to express their rhetorical relation. The cue word also determines which syntactic construction is used to combine the trees. For example, in the last sentence of the example story, the cue phrase *after* introduces a subordinate clause. When Dependency Trees are combined, recurring elements may be deleted. For example, a construction such as *Billy Bones cursed and Billy Bones screamed* will be reduced to *Billy Bones cursed and screamed*.

5.2 Generating referring expressions

An important step in generating a fluent story is the generation of appropriate references to characters and objects. To do this, the Narrator checks (among other things) if the entity being referred to (the referent r) has been recently mentioned, and if there are no other entities of the same gender that have been mentioned equally recently. If both conditions hold, r can be referred to using a pronoun such as *he* or *it*. However, stylistic considerations also play a role: after a pronoun has been used several times in a row, a definite description is preferred to achieve some variation. This is illustrated by the fifth sentence of the example story, which uses *a pirate* where *he* would also have been allowed.

If a regular noun phrase is used, first a noun expressing the type of r is selected (e.g., *pirate* or *ship*). Then the Narrator checks if any adjectives need to be added to the noun, for example to distinguish r from another entity of the same type. Finally, it is decided whether the noun phrase should include a definite or an indefinite article (*the* or *a*). The Narrator generally chooses an indefinite article when r is mentioned for the first time, and a definite article when r has been mentioned before. However, in some cases a definite article can be used at first mention. This is the case with so-called 'bridging descriptions', which refer to an entity that has already been evoked by the mention of another object it is related to. The story in figure 3 contains several examples of this, such as *the hold of his ship*, *the water supply*, and *the deck*. To be able to generate such descriptions, we have defined a number of rules stating that, for example, every entity of type 'ship' has a deck. If r (e.g., a deck) is related to another entity r' (e.g., a ship), the Narrator checks if there is a rule specifying that an entity of the type of r' usually has an entity of the type of r. If there is such a rule, r may be introduced using *the*. In addition, if r' has been mentioned before, and there is no other entity which may stand in the same relation to r (in the example: if there is no other ship the deck could belong to), then the relation between r and r' can be easily inferred and does not need to be mentioned explicitly. For this reason, the example story simply mentions *the deck* and not *the deck of the ship*, since the ship has already been introduced in the second sentence.

5.3 Text and speech

As the last text generation step, the Dependency Trees are converted into actual sentences. Using language-specific knowledge about syntax and morphology, the words in the Dependency Trees are put into the correct order, and nouns, adjectives and verbs are inflected (i.e., suffixes indicating number, tense etc. are added to the word stems in the Dependency Trees). Finally, punctuation is added.

The finished story can be presented to the user in text format, but it can also be presented by a talking face using text-to-speech. Recently, the Narrator has been coupled to DEIRA, a framework for generating spoken reports by an embodied agent [2]. The embodied storyteller can currently produce high-quality speech and lip synchronization; however the generation of appropriate facial expressions is still work in progress.

6 Conclusions and future work

We have described the current state of affairs of the Virtual Storyteller, a framework for the generation of stories based on simulation. We have identified particular issues with the use of autonomous characters for the simulation of stories, and described how the Virtual Storyteller addresses these issues.

Several points remain open for future work. We consider emotions and dialogue to be important aspects of storytelling, but the Character Agents are currently only capable of very simple (reactive) dialogue. Furthermore, the emotion models that we used in earlier prototypes of the Virtual Storyteller are currently under revision. We are investigating what implications story generation has for the modelling of emotion, personality and dialogue, and to what extent this needs to be made particular for a pirate domain.

We continue to explore the role that Character Agents can play as directors of their own stories. One contribution has been the use of framing operators and events to enable plans or justify the adoption of goals.

The Virtual Storyteller is currently capable of generating and presenting simple story texts. As illustrated by our example story, many improvements are still possible at the level of grammar and word choice; however, the real challenge lies in moving from straightforward reports of event sequences to real narratives, employing shifts in perspective, foreshadowing, flashbacks and other dramatic effects which go well beyond the current state-of-the-art in language generation.

Finally, the authoring process for an open-ended story generation project like the Virtual Storyteller remains a difficult issue. We intend to gain more experience in the authoring process by adding more content to our pirate domain, and are currently developing guidelines for dramatic structures that the Plot Agent can aim for by introducing a notion of conflict.

Acknowledgements

This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

[1] Ruth Aylett. Emergent Narrative, Social Immersion and "Storification". In *Proceedings of the 1st International Workshop on Narrative Interaction for Learning Environments*, Edinburgh, 2000.

[2] François Knoppel, Almer S. Tigelaar, Danny Oude Bos, Thijs Alofs, and Zsófia Ruttkay. Trackside DEIRA: A dynamic engaging intelligent reporter agent. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, 2008.

[3] A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

[4] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[5] Ivo Swartjes, Edze Kruizinga, and Mariët Theune. Let's pretend I had a sword: Late commitment in emergent narrative. In *Proceedings of the First Joint Int. Conf. on Interactive Digital Storytelling*, 2008.

[6] Ivo Swartjes and Joost Vromen. Emergent story generation: Lessons from improvisational theater. In *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium*, number FS-07-05 in AAAI Fall Symposium Series, 2007.

[7] Mariët Theune, Sander Rensen, Rieks op den Akker, Dirk Heylen, and Anton Nijholt. Emotional characters for automatic plot creation. In *Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)*, 2004.

[8] Mariët Theune, Nanda Slabbers, and Feikje Hielkema. The Narrator: NLG for digital storytelling. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG'07)*, 2007.

[9] Tom Trabasso, Paul van den Broek, and So Young Suh. Logical necessity and transitivity of causal relations in stories. *Discourse Processes*, 12:1–25, 1989.

Semi-Automatic Ontology Extension in the Maritime Domain

Gerben de Vries[a] Véronique Malaisé[b] Maarten van Someren[a]

Pieter Adriaans[a] Guus Schreiber[b]

[a]*Universiteit van Amsterdam, Human-Computer Studies Lab, Informatics Institute*
[b]*Vrije Universiteit Amsterdam, Business, Web and Multimedia Section, Department of Computer Science*

Abstract

One of the tasks of a maritime safety and security (MSS) system is to map incoming observations in the form of sensor data onto existing maritime domain knowledge. This domain knowledge is modeled in an ontology. The sensor data contains information on ship trajectories, labeled with ship types from this ontology. These ship types are broad and within one type there can be several distinctive behavior patterns. As a consequence we cannot make a good mapping from these trajectories to the ship types. To make this possible we should change the ontology by adding relevant subtypes. This paper presents a semi-automatic method to extend the ontology of ship types on the basis of trajectory data. The first part involves the use of hidden Markov models to model the data of each ship within one ship type and the clustering of these models. The clusters are input to the second part where we use internet querying and natural language processing based ontology extension techniques to extend the maritime domain ontology. We present the promising results of a preliminary experiment that shows an interesting possibility in terms of semi-automatic ontology extension, which would enable an optimal coverage of a given domain: not providing too many concepts, and not leaving essential ones out.

1 Introduction

The goal of maritime safety and security (MSS) systems is to provide their end-users with situational awareness: the ability to detect anomalies in maritime behavior, such as illegal fishing or tankers on a collision course. This is achieved by integrating information from different data sources and processing them intelligently.

One of the tasks in an MSS system is to map incoming observations in the form of sensor data-streams onto existing domain knowledge. Such sensor data-streams provide information about the movement of individual ships. The domain knowledge contains information about the static aspect of ships, it describes ship types and their corresponding characteristics.

More specifically, for the case described in this paper, we have data about ship trajectories on the one hand, and a formal description of ship types and their attributes, i.e. an ontology, on the other. This trajectory data is already labeled with ship types from the ontology, e.g. certain trajectories are from tankers, others from passenger ships, etc. However, these types are broad and do not distinguish all the different behaviors that we observe in the trajectory data. Within one type there can be several distinctive trajectory patterns that might belong to different subtypes. Thus, a concept (ship type) that describes a set of ships that is homogeneous in the existing ontological properties may cover ships with very different trajectory patterns. This means that we cannot classify ships on the basis of their trajectories, we cannot discover inconsistencies between trajectories and other information about ships and we cannot predict their trajectories from their type. In other words: the current ontology does not allow a good mapping between the sensor data and the existing domain knowledge. Therefore, the ontology should be changed.

How to discover possible subtypes on the basis of trajectory data and then to extend the domain ontology with these subtypes is the problem that we solve in this paper. We will show how to learn behavior models for

individual ships, how to cluster these models and how to semi-automatically[1] extend the domain ontology on the basis of these clusters. The result is a combination of machine learning and ontology engineering that is novel and has interesting potential. Machine learning-wise the method is a nice approach to solving the classic unsupervised learning problem of labeling discovered clusters. Furthermore, we do semi-automatic ontology extension based on time-series (e.g. trajectories) data, which is a dynamic and untraditional data source.

The maritime safety and security domain is concerned with events as diverse as the rescuing of pleasure crafts, the monitoring of cargo ship routes or the prevention of smuggling, and objects as diverse as sensors (e.g. radar and sonar), ships, helicopters, off-shore petrol stations, etc. However, the data-set that we work with in this paper is more restricted, it comes from the automatic identification system (AIS). Every commercial ship larger than 300 tons is required to carry an AIS transponder. Such a transponder sends updates about, amongst others: position, heading, speed and rate of turn, every couple of seconds. Furthermore, the AIS specification defines different ship types, which is information that is also regularly communicated, hence our trajectories are labeled with ship types.

In the rest of this paper we first give an overview of the domain ontology that is created on the basis of the AIS specification. Then we will detail the main method in two parts. The first part is based on machine learning and finds possible subtypes by clustering hidden Markov models that model the individual ships' behavior. The second, ontology engineering part, is based on internet querying and natural language processing and, on the basis of the clusters of the first part, expands the domain ontology semi-automatically with new ship types. We will provide an initial evaluation of this method on the case of the 'Passenger Ship'. Finally, we end with a discussion, some conclusions and ideas for future research.

2 The AIS Ontology

To build the ontology modeling the AIS data information, henceforth the AIS ontology, we based ourselves on official documentation describing the different AIS messages: "IALA Guideline No. 1028 On The Automatic Identification System (AIS)"[2]. As the information was contained in tables of PDF documents and as there were very few concepts, with few defining constraints, we modeled this ontology manually, using a standard ontology editor: Protégé[3]. The ontology contains 141 classes and 21 properties, mostly related to the ship types' characteristics. There are only a few ship types: Tankers, Pleasure Crafts, Passenger Ships, High-Speed Craft (HSC), Wing in Grounds (WIG), Special Crafts (with 7 sub-types: Law enforcement vessels, Medical transports, Pilot vessel, Port tender, Search and rescue vessel, Tug, Vessels with anti-pollution facilities), and "Other Types of Ships". Their properties are linked to the fields in the AIS messages: their identifier, speed, length, draught, etc.

The classes of the ontology correspond to the different ship types that we want to distinguish in an MSS context. However, the ontology initially includes no information about trajectories. Taking into account trajectory information may motivate changes to the ontology: for example, a concept that describes a homogeneous set of ships in the original ontology (like Fishing Boat) may cover ships with very different trajectories (for example boats going to fish in the Atlantic ocean will behave differently from local fishermen). This makes it useful to split this concept into sub-concepts with different trajectory types. Therefore, it is mandatory to develop an ontology extension mechanism. As redesigning an ontology is a time-consuming operation, we developed a semi-automatic method, which we will describe in the next sections.

3 Learning and Clustering Ship Behavior

The goal of the method described in this section is to discover possible subtypes of a ship type defined in the AIS ontology on the basis of the trajectories of ships of that type. Trajectory data is an example of multidimensional time-series data. In this time-series data we want to discover clusters (to suggest possible subtypes). The most straight-forward thing to do would be to try to directly cluster the time-series. However, the time-series are very variable in length and they are multidimensional. Moreover, we know very little

[1] The full automatization of this process is still a work in progress at the time of writing of this paper. However, the results will always be validated by a human operator, therefore, the extension mechanism is qualified as semi-automatic.

[2] IALA is the International Association of Marine Aids to Navigation and Lighthouse Authorities.

[3] Protégé is a free, open source ontology editor and knowledge-base framework, available at `http://protege.stanford.edu/`.

about the behavior patterns that we are looking for. Therefore, we have no idea what a good distance measure would be to use for clustering the time-series. In such a case, it makes more sense to use a model based method to compare them, instead of comparing them directly (for a good overview on this, and other issues related to time-series, see [7, Chap. 5]).

The method described below contains three steps. First we preprocess the data, which reduces the size and complexity of the data and produces the required input for the second step. This second step is to train hidden Markov models as 'behavior' models for each individual ship. Thirdly, we define a distance measure on these models and use that to cluster them together with affinity propagation. Although the description of these steps is focused on the case of ship trajectories, the overall method is in principle applicable to any form of time-series data.

The AIS data-set contains trajectory data for a big number of ships, labeled with different types. Let us call a set of ships with the same type: $Ships$. For each $s \in Ships$ we use two time-series: rate of turn, $\lambda^s_{\text{rot},1}, \ldots, \lambda^s_{\text{rot},n}$ and speed over ground, $\lambda^s_{\text{sog},1}, \ldots, \lambda^s_{\text{sog},n}$. These time-series are all sampled at the same fixed sample-rate of ten seconds. We chose these two time-series because they describe the relative behavior of a ship, not the absolute behavior (e.g. this would be the case with position data).

3.1 Preprocessing

To reduce the size of the data-set, the time-series (per ship) are averaged with contiguous, non-overlapping windows of length w. This method is called piecewise aggregate approximation [5]. For a time-series $\lambda_1, \ldots, \lambda_n$, let $\bar{\lambda}_1, \ldots, \bar{\lambda}_m$ be the reduced version defined by[4]:

$$\bar{\lambda}_i = \frac{1}{w} \sum_{j=w(i-1)+1}^{w.i} \lambda_j \ . \tag{1}$$

For our experiments $w = 30$ (300 seconds) strikes a good balance between capturing the general movement of ships and ignoring noise.

By making symbol strings representing the time-series, data is reduced even further. Also, it 'reduces' the multidimensional time-series to one dimension. Moreover, discrete data is the simplest input for hidden Markov models. The first step in this process is to create the set:

$$D = \{(\bar{\lambda}^s_{\text{rot},i}, \bar{\lambda}^s_{\text{sog},i}) | s \in \mathit{Ships} \wedge 1 \leq i \leq m\} \ . \tag{2}$$

Then we make the set D_z by applying z-normalization (see for instance [1, Chap. 9]) to D, i.e. for both attributes we subtract the mean (μ) from each data-point and then divide each data-point by the standard deviation (σ).

The next step is doing standard k-means clustering (see again [1, Chap. 9]) on D_z to get clusters C, i.e.

$$C = \mathit{k\text{-}means}(D_z, k) \ . \tag{3}$$

We use the common Euclidean distance as distance measure. Visual inspection of the clusters showed $k = 5$ to give intuitive results[5]. Each cluster $c \in C$ is assigned a symbol a_c, thereby creating a vector code-book. With this code-book we create a symbol string for each $s \in Ships$ by assigning the symbol $a_{c'}$ to $p_i = (\bar{\lambda}^s_{\text{rot},i}, \bar{\lambda}^s_{\text{sog},i})$, such that

$$\neg \exists c \in C : \mathit{dist}(c, p_i) < \mathit{dist}(c', p_i) \ . \tag{4}$$

Where $dist$ is the distance measure used in k-means clustering, the Euclidean distance in this case, and p is z-normalized with the μ's and σ's that we used for D_z. Let $Ships_{SYM}$ be the total set of symbol strings, one for each ship.

[4] For notational ease we ignore the case when the length of the time-series is not neatly dividable by w and we are thus left with an incomplete last window.

[5] Essentially we have a cluster representing turning left, turning right, stopping, sailing at an average speed and sailing at a fast speed.

3.2 Hidden Markov Models

To capture the general behavior of a ship we use a hidden Markov model (HMM) [8]. These models are tried and tested, relatively easy to understand and use and often used for modeling sequential data, therefore we chose to use them. However, one could use other models than HMMs, e.g. markov chains, conditional random fields, in this step, provided that one can define a distance measure on such models, so that they can be clustered.

We train multiple HMMs $\gamma_{s,1}, \ldots, \gamma_{s,k}$, where $\gamma_{s,i} = (T_{s,i}, E_{s,i}, \pi_{s,i})$, on a set random subsequences Λ_s of each $s \in \mathit{Ships}_{SYM}$, using the standard EM (for HMMs also called Baum-Welch) algorithm. The transition matrix $T_{s,i}$, emission matrix $E_{s,i}$ and initial state distribution $\pi_{s,i}$ are all initialized randomly. The model i, such that

$$\neg\exists\gamma_{s,j} : j \neq i \wedge \sum_{\lambda\in\Lambda_s} P(\lambda|\gamma_{s,j}) > \sum_{\lambda\in\Lambda_s} P(\lambda|\gamma_{s,i}) \tag{5}$$

is selected as the final model for the ship s.

Using a test selection of ships we have identified that the 'elbow' point in the performance increase[6] gained by increasing the number of states in the HMM is around 6 states, with $k = 5$. Depending on the length of the behavior that is relevant, different length subsequences should be used, e.g. to learn a behavioral pattern that is 1 hour in length, one should use subsequences that are significantly longer. We used a length of 6 hours for our experiments. The total length of the number of subsequences can be in the same order as the length of the original sequence, to make sure that training does not take more time than necessary.

3.3 Clustering

To cluster hidden Markov models a distance/dissimilarity measure is required. Rabiner defines a rather simple one in his classic paper [8][7]:

$$\mathit{Dis}_{HMM}(\gamma_1, \gamma_2) = \frac{1}{n}(\log P(\lambda|\gamma_1) - \log P(\lambda|\gamma_2)) \ . \tag{6}$$

Where $\lambda = \lambda_1, \ldots, \lambda_n$ is a sequence generated by model γ_2. To make it usable with standard clustering algorithms this measure can be made symmetric:

$$\mathit{Dis}'_{HMM}(\gamma_1, \gamma_2) = \frac{\mathit{Dis}_{HMM}(\gamma_1, \gamma_2) + \mathit{Dis}_{HMM}(\gamma_2, \gamma_1)}{2} \ . \tag{7}$$

In our experiment we actually take a set of sequences instead of a single one. Furthermore, this set is the same as the set of sequences that we used to train the models. So, the measure that we use is biased towards the data that the models are trained on.

Using the above measure we can compute a dissimilarity matrix for our set of individual ship models. This dissimilarity matrix can then be input to a clustering algorithm to find clusters of models. As a clustering algorithm we took the relatively new, and very fast affinity propagation (AP) [3]. It works by treating data-points as nodes in a network that pass (voting) messages around, thereby determining in a number of iterations what the best exemplars in the network are. Each data-point is linked to an exemplar, thus we have clusters with exemplars as their prototypes. The algorithm can deal with almost all kinds of distance/(dis)similarity measures (even non symmetric ones). It has two parameters to set: the 'dampening factor', which controls the rate of convergence, and the 'preference', which influences the number of clusters that are discovered.

Traditional hierarchical clustering can be used in this step as well and should provide similar results. The output of the above clustering step is input to the ontology extension process, which is described in the next section.

4 Extending the AIS Ontology

The previous step generates ship clusters, within the types from the AIS ontology. Thus, behavior-based clustering provides more fine grained ship types than the ones already present in the ontology. We have to

[6] The performance is measured by the probability of generating the training sequences.

[7] This is essentially an approximation of the Kullback-Leibler divergence, although Rabiner does not mention it there explicitly.

extend the ontology to provide relevant label(s) for the unlabeled class(es). The methodology that we have defined is to extract the possible labels from Internet, based on the population of the unlabeled clusters: we take as input one field of information in the AIS message, the international identifier of a ship (its MMSI[8]) and query automatically online AIS databases[9] providing ship types amongst other information. We then process the resulting pages and extract all possible ship types per cluster. We use the number of times a specific type is mentioned as an indicator of correctness and discard the types occurring only once, because of the noisiness of the data: sometimes MMSI numbers are incorrect and the databases can also contain erroneous information. Once the label is proposed, the corresponding concept has to be integrated at the right place(s) in the ontology. Our semi-automatic ontology extension mechanism provides only suggestions, to be validated by a domain expert, but might provide more than one possibility. Its full automatization is still work in progress.

It consists of two steps: (1) querying the internet for explicit relationships between (a set of) new concept(s) and (a specific set of) the ones from the ontology and (2) extracting from the resulting files the pairs of concepts for which the queries gave at least one hit, with their related number of hits, and propose hierarchies of concepts based on: the extracted pairs, their number of occurrences, the existing hierarchy and information about the general domain. The resulting proposals for extending the ontology can then be validated or modified by the domain expert. These processes can be compared to the general principles of corpus-based ontology building, like in [6] or [2]: finding terms in textual resources, mining relationships between them, proposing these to domain experts and formalizing the validated results. But our method differs on two points: (1) we base ourselves on a "seed ontology" and intend to find only extensions to the core domain model in the textual resources and (2) we do not rely on a manually selected set of texts from the domain, that might be limited or biased by an expert's selection. Mining the relationships between concepts from the internet also implies a bias, but provides a certain objectivity (at least it goes beyond the selection made by a limited number of experts) and coverage of the domain. The two steps of our methodology are detailed below.

4.1 Querying the Internet to Define New Concepts

Search engines like Google or Yahoo! provide a search API[10] to automatically run a set of queries on their large indexes. We generate an automatic queries list with a specific format: besides the fact that they contain one lexical representation of a concept from the existing ontology and one candidate label of the new concept(s), these two also appear in the query in a standard "Hearst pattern". These patterns have been defined in the field of natural language processing for extracting subclasses from plain text corpora [4]. They contain lexical or semantic elements that can be tailored to a specific domain. We used the most productive and the most domain independent of these patterns: "*(X) such as (SuperClassOfX)*" and "*(X) and other (SuperClassOfX)*", where *X* is the new concept and *SuperClassOfX* is a possible superclass. As these patterns require also the plural form of the label of the classes, we used the Celex lexicon[11] to compute these, and built all possible queries (queries including the singular or plural form of all the concept's labels concerned, in both of these patterns). So, in our case *X* represents any textual representation of the new concept and *SuperClassOfX* any textual form of the different ship types already present in the AIS ontology, which would then correspond to a superclass suggestion for the new concept.

4.2 Proposing New Hierarchies

The pairs of terms for which the queries provided at least one hit are extracted from the results' file (so the new concept and another ship type), along with their number of hits (corresponding to their number of occurrences on the internet), as a measure of relevance for the relationship between the concepts of the pair. This number, the analysis of the existing hierarchies in the AIS ontology (the fact that some ship types are already modeled as subclasses of each other) and some domain knowledge (like the fact that a Ship and a Craft can be considered as subclasses, according to Merriam Webster's[12] online definition: "Ship: a large

[8] MMSI stands for Maritime Mobile Service Identity.

[9] We queried the Google database at `http://aprs.fi/info/` and the one from the International Telecommunication Union at `http://www.itu.int/cgi-bin/htsh/mars/ship_detail.sh?`.

[10] `http://code.google.com/apis/ajaxsearch/documentation/` and `http://developer.yahoo.com/search/`

[11] `http://www.ru.nl/celex/`.

[12] `http://www.merriam-webster.com/thesaurus/ship`

craft for travel by water") are used to propose possible hierarchies for extending the ontology, and rank the propositions. The ranking procedure is not automatized yet, but follows these heuristics: rank the pairs of terms based on the number of hits of the Hearst patterns corresponding to their lexical representations, check the existing hierarchy in the ontology to find subclasses within the proposed possible super-classes of the new concept, use domain knowledge to find synonyms or further sub-classes between them and propose the resulting hierarchies' leaves as ontology extension suggestions.

5 Experiment

To test the feasibility of the above method we have set up a preliminary experiment using the AIS data-set.

5.1 Clustering Passenger Ships

In the data it occurs a number of times that ships stop sending AIS signals, for instance because they are in port or leave the area of coverage. If the time difference between the last signal and the new signal is significantly long (e.g. fifteen minutes), then we treat this signal as if it is the start of a new sequence, i.e. one ship can have a *set* of time-series.

From the AIS data we have selected all the ships that were labeled 'Passenger Ship' according to the AIS ontology. This data was first preprocessed as described earlier. For each ship we trained 10 hidden Markov models, with 6 states. We then selected the one with highest performance, such as defined in (5), for each ship. Using the dissimilarity measure (7) we computed a dissimilarity matrix as input for clustering with affinity propagation (AP). One of the parameters of AP, the 'preference', depends on the maximum dissimilarity in the dissimilarity matrix, and it requires some tuning by hand. In this experiment -12 worked fine, but unfortunately, as said, it completely depends on the dissimilarity matrix. The 'dampening factor' is less crucial and is fine on the standard setting of 0.95.

The result was two clusters. When we visually inspected the trajectories of the ships in the different clusters we saw that they were clearly distinct. One cluster showed the pattern of ships moving between two harbors regularly, with short intervals. The other cluster contained trajectories that had ships sailing in long, more or less straight, lines. The contents of these clusters are passed on to the next step.

5.2 Extending the AIS Ontology

5.2.1 Mining possible labels from the internet

As mentioned previously, this operation is performed in two steps: searching the internet for possible labels for the two ships' clusters and processing the resulting pages to isolate this precise information: the labels. We queried the aforementioned AIS databases with the MMSIs of the ships from the two different clusters and processed the resulting pages (fetched and stored locally): we extracted the ship types that were mentioned for each successful query.

Results for cluster number one This cluster contains 12 ships, for 9 of them the databases gave a type result: Passenger Ship "general" (4 times), Passenger Ship carrying hazard/pollutant cat A (3 times), Passenger Ship carrying hazard/pollutant cat C (once) and Ferry boat (once). Given the fact that we want to cluster ship types based on their behavior, no relevant sub-class of Passenger Ship should be modeled based on the first three types: a Passenger Ship carrying a specific cargo is dealt with as a property attached to the generic Passenger Ship class, this specification does not require a specific subclass in the ontology. The concept of Ferry, however, has a very specific behavior amongst the Passenger Ships and this concept is not present in the ontology yet. This would be a necessary extension to the ontology, if the Ferry label would represent a (whole and) separate cluster. However, the behavior of this particular ship was not distinguished from the ones of the generic Passenger Ship type by the machine learning process. Given the fact that there is also only one single instance of this specific type, in an otherwise coherent cluster, and given the fact that real-life data is often noisy, we did not create the extension for this single case.

Results for cluster number two We got results for 5 out of the 6 ships of the second cluster, but for 3 ships the information given by one of the databases was merely "Not available". For the same 3 cases, the "Ship Class" provided by the other database was "MM FBT", which means Merchant Ship - Ferry Boat. For one of

the other 2 ships, the MMSI number was erroneous (a default number), and the last one was classified only as "Passenger Ship 'general"'. We decided to use the information given by 3 of the 6 ship's identifiers, and consider that the second cluster represented Ferries. An observation of the trajectories of the corresponding ships, and the additional knowledge about their starting and end points (ports linked together by a Ferry line) confirmed the validity of this intuition. We consider the one labeled "Passenger Ship 'general"' to be noise.

We had a closer look at the Ferry from the first cluster, to get insight about why it had not been clustered with the others. It turned out that this ferryboat was linking harbors from two different countries; our AIS data concerns only the information sent in one of these two national waters, thus we had only partial information about that ship's trajectory: the messages were stopped when the ship entered the second national waters, and we could not detect the typical "ferrying" pattern.

Once the Ferry's label is elicited, we wanted to have the corresponding concept semi-automatically integrated in the ontology, in order to update our domain knowledge. The method that we have employed for this is presented in the next section.

5.2.2 Analyzing the results and updating the ontology

We created automatically (by Perl scripts) a list of queries for the Yahoo! Search API, and stored the XML results locally. We processed the results and extracted from the files:

- the pairs of terms in a sub-class relationships for which the query had provided hits,
- the number of possible results (the number of hits that Yahoo! found in its index) and
- the number of retrieved results (as common internet users do not usually go beyond the third result page, we had set as threshold to retrieve at most 30 hits, so the actual number of pages is comprised between 0 and 30).

We got 2309 potential results (number of hits on Yahoo!'s index of the internet) and 262 actual ones (really retrieved pages, according to our threshold of 30 pages at maximum). The couples of concepts from the AIS ontology that were found on the Internet in a superclass relationship with the concept of Ferry were: Ships (2085 possible hits), Crafts (171), Passenger Ships (29), Other Types Of Ships (15) and Pleasure Crafts (7). Tankers and Fishing boats were not returned by our methodology, although the latter is often in close co-occurrence with one of the possible labels of the concept of Ferry on Internet. This shows the advantage of using our method over using a similarity based on simple co-occurrences in the internet in general (a principle that inspires, for example, the Google distance score between two terms): it gives more precise results. We intend to create hierarchies as specific as possible. Lexical inclusion indices, as investigated in [9], show us that Pleasure Craft have good chances to be a subclass of Crafts because "Crafts" is included in the label "Pleasure Crafts" and because the grammatical category (or Part Of Speech (POS) tag) of the extension is an adjective. In the same fashion, Passenger Ships could be a subclass of Ships, but with lesser certainty because this time the POS tag of the extension is a noun.

Nevertheless, as a first approximation, we propose the following hierarchies: Ships $>$ Passengers Ships $>$ Ferries, Crafts $>$ Pleasure Crafts $>$ Ferries and Other Types Of Ships $>$ Ferries. Pleasure Crafts, Passenger Ships and Other Types of Ships are already present in the ontology, as subclasses of Ship Types, so we suggest only the last level of these hierarchies as extensions. Ferries, according to what we found on Internet and to the core domain knowledge that we had, can either be a Passenger Ship, a Pleasure Craft or an Other Type of Ship. Considering the number of hits per possibility, and the fact that all the ships from the second cluster are all initially labeled with the broad category Passenger Ship, we give a higher rank to this suggestion. This suggestion has, however, to be validated by a domain expert, who can choose to keep all three possibilities, just two or decide for another one.

We still have to automatize this ranking and develop an interface for the interaction with the domain expert to update the AIS ontology directly. This automatization is currently in progress.

6 Discussion, Conclusion and Future Work

We have presented a method to semi-automatically extend a maritime domain ontology on the basis of trajectory data. It discovers possible subtypes of an existing ship type by finding clusters of behavioral patterns in trajectories for that type. Based on these clusters we use internet querying and natural language processing techniques to discover labels, and their place in the ontology, for the suggested subtypes.

From the perspective of machine learning it is a nice approach to solving the ever existing unsupervised learning problem of labeling discovered clusters. In ontology engineering the idea of semi-automatic ontology extension from the point of time-series data is rather novel. The presented method holds promise for the future: the combination of machine learning and ontology engineering works well and this application shows that there are interesting possibilities to be explored.

However, it is clear that more experimental evaluation is required. Not only do we need to do this in the maritime domain, but it would also be useful to test the methods on other comparable domains, the first that come to mind are other domains with moving objects, such as cars and planes. In principle the methods can be applied to any domain with dynamic, time-series data on the one hand and static, ontological knowledge on the other hand.

Another problem is that, for now, we need good information about the specific instances of the domain to be available on the internet. It is a challenge to find other properties of the clusters (i.e. not the specific instances) that we can use for the internet search.

Also, it would be nice if clustering of the entire data-set could be the basis of the ontology, instead of using a hand-crafted "seed"-ontology as we do currently (e.g. this would be useful in the case of completely unlabeled trajectory data). However, at least with the AIS set, the current approach is not able to do that well. For one thing because it is very slow, since it requires the computation of a large dissimilarity matrix. Furthermore, the currently used AIS attributes (rate of turn and speed over ground) do not seem to provide enough discriminatory information.

The hidden Markov model as the basis for behavior models was chosen because it is well-known and often used, however, this is not to say that there might not be models that represent behavior more accurately. Furthermore, the current training of the HMMs, based on random initialization, is rather slow, for other models we might be able to this faster.

Apart from these issues, we have more ideas for future work. First of all, we will look at using more data sources: other kinds of sensor data streams (for instance radar), but also other types of sources of domain knowledge. Currently there is no feedback from the extended ontology back in to the machine learning. Potentially this can be very interesting, because with the new labels we can start the process again and extend the ontology even further. An interesting issue here would be when to stop this loop to get optimal coverage of the domain: not providing too many concepts, and not leaving essential ones out.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] Brigitte Bibow and Sylvie Szulman. *Knowledge Acquisition, Modeling and Management*, volume 1621/1999 of *Lecture Notes in Computer Science*, chapter TERMINAE: A Linguistics-Based Tool for the Building of a Domain Ontology, pages 49–66. Springer Berlin / Heidelberg, 1999.

[3] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[4] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[5] Eamonn J. Keogh, Kaushik Chakrabarti, Sharad Mehrotra, and Michael J. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *SIGMOD Conference*, 2001.

[6] Alexander Maedche and Steffen Staab. Kaon - the karlsruhe ontology and semantic web meta project. *KI*, 17(3):27–, 2003.

[7] Fabian Mörchen. *Time Series Knowledge Mining*. PhD thesis, Philipps-University Marburg, 2006.

[8] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[9] Eric SanJuan, James Dowdall, Fidelia Ibekwe-Sanjuan, and Fabio Rinaldi. A symbolic approach to automatic multiword term structuring. *Computer Speech & Language*, 19(4):524–542, 2005.

The Effects of Cooperative Agent Behavior on Human Cooperativeness

Arlette van Wissen Jurriaan van Diggelen Virginia Dignum

Institute of Information and Computing Sciences
Utrecht University, the Netherlands
arlette.vanwissen@phil.uu.nl; {jurriaan,virginia}@cs.uu.nl

Abstract

This paper explores the question of how the cooperativeness of a software agent affects the cooperativeness of a human player. We implemented three kinds of agent behavior to examine human cooperativeness. In our experiment we used the Colored Trails game as a negotiation environment. Our data shows that humans behave more cooperatively towards agents that negotiate with them in a cooperative way and that humans tend to punish egoistic and unfair play by behaving non-cooperatively themselves. We also find that egoistic agent behavior is often mistaken for human behavior.

1 Introduction

In this paper we focus on human-agent interaction and in particular on the question of how humans treat a computerized opponent and in what respect this is different from how they would treat a human opponent. On the one hand, human players tend to be cooperative towards their human opponents but punish other human opponents severely in case of unfairness. On the other hand, research has shown that there is a significant difference between the reaction of people to automated opponents than to human opponents [12]. This indicates that humans perceive other people differently from computerized opponents. It remains an open issue how these two phenomena are combined when humans interact with software agents and what kind of effect knowledge that the opponent is an agent has on human behavior. It might be the case that humans stick to their fair play and helpful manner. It could also be that they will exploit more altruistic agents for their own benefit. We are interested in studying the cooperative aspects of human behavior in relation to the agent and we will examine how this can be altered by implementing several behavioral strategies of the computerized opponent. The central question addressed in this paper is: how does the cooperativeness of a software agent affect the cooperativeness of a human player?

In order to explore this question, we must understand how humans respond to agents with conflicting interests and the way in which people react to different conducts of agents. Several prior studies revealed interesting properties of human negotiation behavior. For example, human reasoning and negotiation strategies are affected by social factors, such as altruism, fairness and reciprocity [2]. Furthermore, people seem to use strategies with social preferences that do not correspond with the predictions of standard economic game theory: humans may not be completely rational reasoners [2, 9, 7].

To analyze differences in behavior between various kinds of human-agent interactions we used Colored Trails [7], which is an environment that enables us to design and evaluate decision-making behavior and dynamics between players. Colored Trails is a conceptually simple but strategically complex game that provides such a testbed. This game has been developed for the purpose of testing strategies for automated agents that operate in groups. These groups can include people as well as computer agents. Players have to exchange resources in order to reach their goal. Some degree of cooperativeness is therefore essential.

We used a version of the Colored Trails game to implement agent strategies that varied in their degree of cooperativeness. In some conditions subjects were not aware of the identity of their opponent, i.e. they did not know whether they played against a human or an agent. In this way, we were able to analyze the relation between the cooperativeness of agents and that of humans. We tested two main hypotheses: people will play cooperative towards an altruistic opponent and they will punish egoistic behavior. Our results support these hypotheses and showed three main findings. First, humans tend to be more cooperative towards a cooperative

opponent and do not fully exploit altruistic behavior. Second, humans seem to play less cooperative towards egoistic opponents. Third, an egoistic opponent is often perceived as a human.
The results of our research contribute to the design of interactive systems that encourage and facilitate human cooperativeness and altruistic behavior. Such behavior is for example necessary in crisis management situations that require a quick response by people based on information or requests received from an artificial team member, such as a (PDA-based) personal assistant. In such situations, it is desired that the human team member reacts quickly and without reserves to possibly unexpected requests.

The paper is organized as follows. First, we will discuss related work and show how our work differs from it. The conceptual approach to our experiment is presented in Section 3. This section is mainly concerned with introducing the Colored Trails game framework and demonstrating how we use it in this project. In Section 4 the experimental setting will be addressed. The results of the experiments will be discussed in Section 5, followed by our conclusion in Section 6.

2 Related Work

Several studies have tried to develop models for agents that negotiate with humans. For instance, [11] examines how people treat human opponents and computerized opponents when they design and implement an agent for negotiating on their behalf. As it turns out, programmed agents show no difference in conduct toward agents and human uncooperative opponents. Ficci and Pfeffer [3] introduce a model of human reasoning which shows that players form slightly incorrect beliefs about each other's preferences. A recent study of [4] modeled agents that use social preferences to predict individual bids of people. Their results seem to show that people have a tendency to be more generous than necessary in order to establish cooperation. On top of this, the negotiation experiments presented in [6] seem to imply that human responders care about equality of outcomes while negotiating with agents.

None of these prior works have investigated the effects of different computational strategies on human behavior in a repeated interaction scenario. The work of Gal et al. on reciprocity [5] does evaluate computer models of human behavior in repeated interaction but does not position computer players against people. In this project we will combine and extend existing work to understand the effect of agent behavior on the cooperativeness of people towards their opponents. On the one hand, we will develop agent behavior models that exhibit different cooperativeness strategies, differing from pure egoism to pure altruism. In this sense, our work is similar to that of e.g., Gal et al. [6] and Kraus and Grosz [7]. On the other hand, we are interested in checking whether results of studies on mimic behavior of people towards others also cover their interactions with agents.

Instead of implementing a score-dependency to encourage cooperation (as is done in [3, 6, 11]), we developed an environment that stimulates cooperation by implementing an environment based on an iterated Ultimatum Game. The Ultimatum Game [8] is a commonly used game setting to simulate and analyze negotiation behavior. In the classic ultimatum game, two people are given the task to divide a sum of money. One player proposes how to divide the money between them and the other player can accept or reject this proposal. If the second player rejects, neither player receives anything. In the original game, the players interact only once. As it turns out, people offer 'fair' (i.e., 50:50) splits, and offers of less than 20% are often rejected. This latter phenomenon is often referred to as *inequity aversion*.

3 Experimental Approach

Cooperative behavior is a broadly discussed and controversial subject, since there are many theories about the causes and ultimate motives of (non-)cooperative behavior [10]. Our concern lies with the effect of agent behavior on human behavior; we do not aim to give any final explanation of underlying motives for cooperation. We use the word *cooperative* for interacting together willingly for a common purpose or benefit. This behavior is said to be *altruistic* if it involves a fitness cost to the proposer and confers a fitness benefit on the responder. It is said to be *egoistic* if the the proposer acts only to benefit himself by increasing his fitness.[1] We use the notion of *helpful behavior* to refer to cooperation that does not necessarily imply a fitness cost for the proposer, but in any case yields a fitness benefit for the responder. *Reciprocal* behavior

[1] Notice that we use the *evolutionary* notions of egoism and altruism here and disregard the *psychological* notions. For an account of the evolution and psychology of (un)selfish behavior, see [13].

are conducts that correspond to the behavior of others. In this study, we created a setting that encourages and instigates these different kinds of cooperation.

We have developed an experiment for investigating the effect of agent behavior on human cooperativeness towards artificial agents. We use an existing agent-based negotiation testbed, Colored Trails, as basis for interaction between humans and artificial agents and developed different agents to simulate extreme interaction behaviors, both cooperative and non-cooperative. In the remainder of this section, we describe the experimental approach in detail.

3.1 Colored Trails

Colored Trails (CT) is a game developed by Grosz and Kraus [7] which provides a testbed for investigating interactions between players who's task it is to reach a goal by exchanging resources. Computational strategies and human decision making can be studied by comparing interactions in both homogeneous and heterogeneous groups of people and computer systems. The game is very suitable for our purpose, since we want to measure possible influences of agents on human behavior. It is played on an N×M board of colored squares and one or more squares can be designated as a goal. Each player has to reach the goal by exchanging chips. In order to move a player piece along the board, the player has to hand in a chip with the color corresponding to the color of the adjacent square he wants to move to.

There are several ways in which the game can promote inter-dependency of players. If players are *task dependent*, they are depending on other players supplying their chips for the performance of their task (reaching the goal). Moreover, this is a social dependency because the score of one player depends on the helpful behavior of his opponent. Another dependency can be created by making the players *reward dependent*. This indicates that the score of one player depends in a way on the score of the other players. In this way, the competitive relation between the proposer and the responder can be strengthened or weakened by the scoring function.

3.2 Conceptual Design

We used the CT framework to implement a negotiation environment that is similar to an iterated Ultimatum Game (UG). The stable outcome of an UG where each player has chosen a strategy and no player will benefit by changing his strategy (under the assumption that the other players keep theirs unchanged), is called the Nash Equilibrium. In this case a Nash equilibrium would occur when the proposer offers the smallest possible amount of chips to the responder and the responder accepts. However, experimental evidence shows that the proposer offers a relatively large share to his opponent and that the responder often rejects smaller positive amounts [8]. This can be interpreted as human willingness to play fair and to punish 'unfair' splits. The results of Gal et al. in [6] seem to be coherent with this scenario. We use the UG to test whether humans show similar helpful or punishing behavior when they interact with a software agent. Our scenario is more complex that the UG, because it involves decisions within a task setting, rather than just a list of choices (each choice representing a different division strategy).

The game consists of several rounds in which two players have alternating roles: *proposer* or *responder*. Each round, different phases define the action that can be undertaken by the player. During the communication phase, the proposer creates a proposal to exchange chips with his opponent. The opponent can either accept or reject this proposal. The proposer can also decide not to exchange any chips. In case of rejection, both players receive nothing. This means it is not possible to negotiate in order to collect chips over several rounds or to adjust the proposal. This is defined as *one-shot negotiation*. The responder determines the final chip distribution of a round by accepting or rejecting the proposal. During each round, both players have to try to obtain all the chips they need in order to reach the goal. We used the CT environment to create the following protocol of a round:

1. **orientation phase** The player is able to get accustomed to the board and its new chips. During this phase, communication is not possible.

2. **communication phase** Both players have to try to obtain all the chips they need in order to reach the goal. The proposer can offer one proposal and the responder can react to it.

3. **redistribution phase** Agreements are enforced by the game controller (which is implemented in CT): after the proposal is accepted or rejected, the game controller will redistribute the chips according to

the proposal. This phase gives the player the opportunity to observe the changes that have been made to the chipsets.

4. **movement phase** If the player has the necessary chips to reach the goal, the program will execute the player's movement by moving its piece to the goal via the shortest possible path. If on the other hand the player does not have the required chips, his piece will not move at all. Again, we use an all-or-none approach to stimulate cooperation.

The scoring function in this experiment does not stipulate a reward dependence, but only a task dependence. Players did not receive penalties for the amount of chips they had left at the end of a round and their score was in no way dependent of that of the opponent. By making their score independent, we eliminated the fuzzy kind of altruism that might occur when one player helps the other player to reach the goal. This might still be regarded as egoism, since the helping player benefits from his opponent reaching the goal[2]. In our scenario, any helpful act from a proposer or responder would be to make more 'fair' or perhaps even altruistic proposals. Cooperation is stimulated by playing an iterated game. Participants have the prospect of encountering their opponent again in the game and are therefore more inclined to cooperate [1].

3.3 Agent Models

We developed three different kinds of agents: the altruistic agent (ALT), the egoistic agent (EGO), and the reciprocal agent (REC). The behavior of these agents is quite extreme and prototypical. It is important to stress here we did not try to make agents that would act as much like humans as possible. We wanted to observe the differences in human behavior that is caused by agent behavior. For that purpose, we designed agents with very different utility functions. ALT is satisfied with a suboptimal path to reach its goal and will grant almost all requests, thereby creating a good reputation for itself. EGO has a very negative strategy: it will not do the opponent any favors and always aims for the shortest path.

Table 1: The different agents and their strategies

Egoistic Agent (EGO)	**proposer**	*offers*	randomly: a) no chips c) b) chips that are not beneficial for opponent
		requests	a) chips that it needs to reach the goal b) chips that enable shorter path
	responder	*accepts*	deals with better score than it can obtain with current chipset
		rejects	all other deals
Altruistic Agent (ALT)	**proposer**	*offers*	chips unrequired to reach goal that are useful for opponent
		requests	chips it needs for a path to goal with least costs for responder
	responder	*accepts*	a) deals in favor of itself b) deals in favor of opponent if it can already reach its own goal
		rejects	deals that make it unable to reach the goal
Reciprocal Agent (REC)	**proposer**		a) behaves as egoïstic proposer if favor balance =<0 b) behaves as altruïstic proposer if favor balance >0
	responder		a) behaves as egoïstic responder if favor balance =<0 b) behaves as altruïstic responder if favor balance >0

REC has a mildly adaptive strategy that is a combination of the strategies of EGO and ALT. It uses a *favor balance* that keeps track of how cooperative its opponent has been towards it in the past. The favor balance is calculated in the following way: *fb = 1 + pos_encounters - neg_encounters*. The default action of REC is to act altruistically, which is ensured by adding 1 to the favor balance. In case a proposal or response is judged as noncooperative, the number of *neg_encounters* is increased, in case it is considered positive, it will increase *pos_encounters*. A response of the opponent is considered negative if it is a rejection and is judged positive if it is an acceptance. A proposal of the opponent is considered negative if it would leave the agent with a less advantageous chipset than it had beforehand, and is judged positive otherwise. Table 1 gives an overview of the three agent strategies.

We implemented these strategies to explore two hypotheses. Both hypotheses have as a starting point that even though research has shown agents to be perceived differently than humans, the human tendency to play fair and to enforce others to do the same will dominate.

[2]This is often referred to as *evolutionary egoism*. Both evolutionary egoism and *psychological egoism* (helping others because it makes you feel good) can encourage cooperation. For an in depth discussion, see [13]

Hypothesis 1 Cooperative behavior of the agent instigates cooperative behavior of the human opponent. More specifically, both the altruistic and reciprocal agent, even though the altruistic agent is vulnerable to exploitation, will receive helpful behavior from their opponent.

Hypothesis 2 Egoistic (and therefore non-cooperative behavior) instigates 'punishment-behavior' of humans. We expect human players to offer the egoistic agent as little as possible and to prevent the agent from reaching its goal.

4 Experimental design

4.1 CT Configuration

CT makes use of a wide variety of parameters that allow for increasing complexity of the game along different dimensions. Our games were played with *full board visibility* and *full chip visibility*: both players had complete knowledge of the board, both their positions on the board and the chip distribution. Since we used one-shot negotiation games, full knowledge was necessary to make appropriate proposals and responses. Their full visibility allowed players to make deliberate decisions about helping their opponent or not. The PathFinder is an additional tool for players that shows the chips needed to take the shortest path to the goal. In this experiment, players were only allowed to see their own chips in the PathFinder and PathFinder did not show any paths longer than the shortest one.

Figure 1 shows our basic CT configuration. The game is played on a 4×5 board with one square designated as a goal. The scoring function, board configuration and positions of the players and the goal on the board remain the same throughout the game. The player to start and the role he[3] performs (proposer or responder) are determined randomly. Each round, the game manager randomly allocates one of ten different sets of chipsets to the players. These chipsets are constructed in such a way that 'interesting' negotiation behavior could follow from it: there are always enough chips to allow both players to reach the goal. Yet initially the chips are thus distributed that it is not possible for both players to reach the goal via the shortest path.

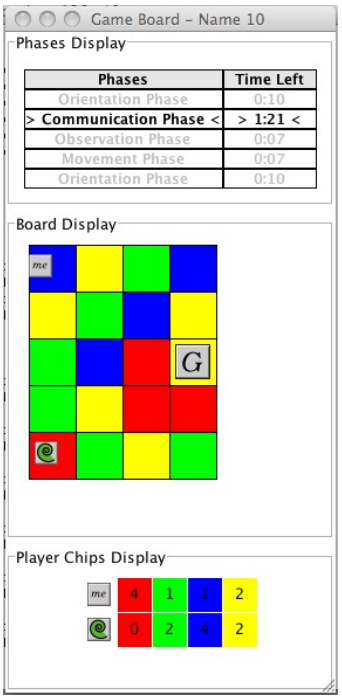


Figure 1: The configuration of the CT game showing the board, the phases of a round and the players' chipsets

4.2 Setup

A total of 30 subjects participated in the experiment. They were divided into 4- or 6-player groups. All participants were graduate and undergraduate students between 20-30 years of age. 80% of them studied Artificial Intelligence, 20% were enrolled in other studies. Almost 50% of the participants were women. Each experiment took two hours and started with an instruction and a tutorial game. Subjects could not observe the terminals of other subjects. Each subject participated in 4 games, adding up to a total of 120 games played. Participants were instructed to perform a non-competitive task: they were to try to maximize their own scores, not to minimize other players' scores. This is also reflected in the score function, since the majority of the points can be received by reaching the goal, not by choosing the shortest path. It follows that there is no strictly competitive or zero-sum iteration. During the instruction, subjects were told how the scoring function was set, but their score was only displayed at the end of every game. We restricted the score display to demotivate the participants to compute relative score improvements. A financial bonus payment would be rewarded to the player who obtained the highest score. The scoring function was defined

[3]Both male and female subjects as well gender-neutral agents were involved our experiment. For purposes of convenience and clarity we will refer here to all subjects as male. This is however completely arbitrary.

as follows: *s* = *goal* + *board_size* - *taken_steps*, with the following variables:

goal	rewarded points for reaching the goal	100
board_size	size of the board	20
taken_steps	number of steps of the taken path	variable per round

Considering our main goal was to examine the differences in human behavior when confronted with (non-)cooperative behavior of an agent, we had to compare this behavior in some way to the behavior of humans towards a human opponent. For this reason, participants played three heterogeneous games against computerized opponent and one homogeneous game against another human participant. The group that played the human condition was used as a control group to demonstrate the degree of cooperativeness of the players. For these experiments, we hypothesized that human inequity aversion would dominate their indifference to agent strategies. We expected that telling the participants they would be playing against computer agents could alter their behavior. In order to investigate the effect of knowledge and beliefs of their opponent on the acceptance level of 'unfair' or noncooperative behavior, we distinguished their knowledge and beliefs about the opponent from their behavior. Hence the general setup included four variables:

1. **Nature** of the player: human or agent.
2. **Strategy** of the agent: altruist, egoist or reciprocal.
3. **Knowledge** of the nature of opponent: True Belief (TB), False Belief (FB), No Belief (NB).
4. **Order** of played opponents: egoist, altruist, reciprocal, human.

We created three groups of ten participants each and combined them with different belief conditions. The belief condition provided the participants only with information about the nature of the opponent; the strategy of the agents was never disclosed. The game controller randomly determined the order of the opponents for each of the three groups. During the game, subjects communicated trough CT's GUI and they could not see each other's terminal. Table 2 shows the three conditions for the three groups. Each game consisted of ten rounds. In the TB-condition we told the participants the truth about when they were playing against an agent or against another human. In the FB-condition we misled them by announcing their opponent would be a human when in fact it was an agent, and the other way around. Unlike the other conditions, we did not give the players any information at all about their opponent in the NB-condition.

4.3 Evaluation

At the end of each game players were asked to fill out a questionnaire. This questionnaire provided us with insights of how different aspects of cooperation, such as altruism or egoism, come about. First of all, subjects had to write down what they believed was the nature of their opponent. Second, we asked them to describe their own strategy and inquired how cooperatively they perceived themselves. Finally they had to answer some questions about the strategy of the opponent and its degree of cooperativeness. We used this information to find out how the participants perceived their opponents and how this influenced their cooperative behavior.

Since human perception and self reflection may differ from reality, we also saved the logs of each CT game. They contained all the vital information of the game, such as score, the proposals and responses made, whether both players reached their goal and if so, how many steps it took. These logs provided us with information of how often players made fair trades or even helped their opponents.

Table 2: Overview of the rounds played against different opponents under various knowledge-conditions

	Egoistic Agent	Altruistic Agent	Reciprocal Agent	Human
True Knowledge	10	10	10	10
No Knowledge	10	10	10	10
False Knowledge	10	10	10	10

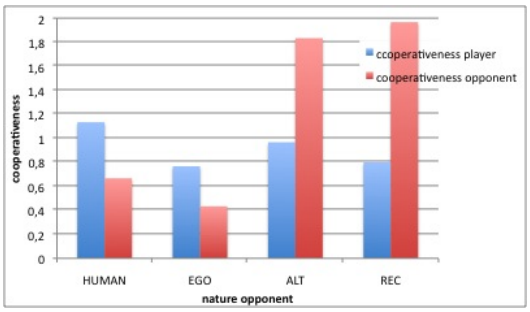


Figure 2: Results of how cooperative humans perceived themselves and their opponents. Scale 0 - 2: 0 = non cooperative, 1 = partially cooperative, 2 = very cooperative

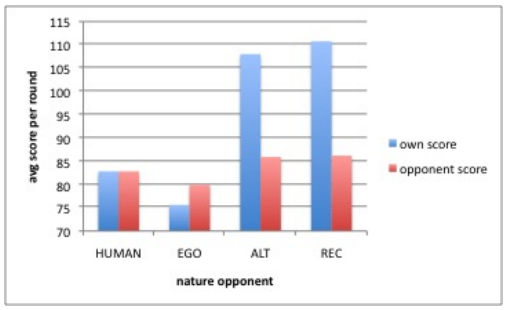


Figure 3: Average score per round

5 Results

We expected humans to play more cooperative towards opponents that behave cooperatively or altruistically themselves (H0) and we hypothesized that egoistic behavior would be disciplined (H1). Our results seem to support both hypotheses[4]. More extensive research has to be done to significantly demonstrate these results. Firstly, the questionnaire shows that the agents' strategies were correctly identified: 100% of the participants found ALT moderately to very cooperative: 83% found it very cooperative and 17% found it partially cooperative. Interestingly, REC was also considered very cooperative: 97% of the subjects perceived it as cooperative. This can be explained by the fact that the first action REC takes is an altruistic one, it only becomes egoistic when the opponent treats it in an egoistic way. 63% found the egoistic agent not cooperative at all but surprisingly almost half of the subjects believed the agent to be partially cooperative (47%). This is even a larger percentage than the amount of subject that found their human opponent cooperative (40%). As it turned out, participants judged the agent as being cooperative because it sometimes offered chips when it did not need anything in return. It did not seem to matter that these chips were of no use to the responder.

Results show that human participants were more cooperative towards opponents perceived to be cooperative. Figure 2 reveals that the cooperativeness of human players increases as their opponent shows more cooperative or altruistic behavior. This result is most clear when we compare the degree of cooperativeness towards altruistic and egoistic agents, respectively 0.97 and 0.77 on our two point scale. Furthermore, players consistently identified themselves as more cooperative than their (human) opponents.

Figure 3 shows the average score of subjects per round, playing against different opponents. I.e. the category 'ALT' refers to the experimental condition of the participants playing against an altruistic opponent. In this case, the categories show the average of all belief conditions (True Belief, False Belief and No Belief). Without any form of cooperation, the players would be able to obtain 70 points per round. In our experiment, the altruistic and reciprocal agent have an average score of 85,9 and 86,2 respectively. The egoistic agent remains behind with an average score of 80. Figure 3 shows the following main results:

1. The score of the agent increases if it has a more cooperative strategy.
2. Human cooperation with the egoistic agent does pay off, because the average score of both the human player and the agent transcend the minimum score of 70.

Results indicate that the score of the agent increases if it behaves cooperatively and it decreases in case of egoistic behavior. Interestingly enough, it appears that people do not fully exploit altruistic players. The logs show that altruistic agents reach their goal slightly more often (75% of the time) than both human (73% of the time) and egoistic players (69% of the time). The questionnaires showed that participants were prepared to give the altruist chips that would help it to reach its goal.

Finally, in different game conditions (cf. table 2) players received no, false or true information about the nature of their opponents. At the end of the game, they were asked to identify their opponent as a human or an agent. The results can be found in Table 3. Remarkably, the egoistic agent was correctly identified only at chance level, i.e. in 50% of the cases they are mistaken for human. The results of the questionnaire clearly show that the behavior of the altruistic agent is seen as 'stupid', 'dumb' and 'too cooperative to be human'. A possible explanation might be that people expect other people to behave selfish and egoistic in

[4]For true statistical significance, results need to be confirmed with a larger number of participants.

Table 3: The percentage of human players that correctly guessed the nature of their opponent in that condition.

Nature of opponent	% players that correctly guessed opponent's nature
HUMAN	63%
EGO	50%
ALT	83%
REC	83%

negotiations. This seems to correspond to the findings of Kraus and Grosz in [7], who suggest that system designers should build cooperative agents because people exhibit and expect agents to be cooperative.

6 Conclusion

In this paper we have explored the question of how the behavior of a software agent affects the cooperativeness of its human opponent in a negotiation setting. We used the CT game as a negotiation environment and implemented three kinds of agent behavior to examine human cooperativeness. Initial results show that humans behave more cooperatively towards agents that negotiate with them in a cooperative way. We also find that humans tend to punish egoistic and unfair play by behaving non-cooperative themselves. Furthermore, egoistic agents are more likely to be identified as humans than altruistic agents, which are often perceived as 'dumb'. Although our experiment is small in scale, our results suggest that both humans and agents benefit from a cooperative strategy. In order to achieve statistical significance of the results, in the future, we will extend our experiment to a larger number of participants. We plan to develop the agents' behaviors to more complex ones, in order to further explore the conditions under which cooperative behavior emerges.

Acknowledgements. We thank Ya'akov Gal and Bart Kamphorst for helpful comments and assistance with the Colored Trails system. This research is funded by the Netherlands Organization for Scientific Research (NWO), through Veni-grant 639.021.509.

References

[1] K. Binmore. *Fun and Games: a Text on Game Theory*. D.C. Heath and Company, 1992.

[2] C. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princ. Univ. Press, 2003.

[3] S. Ficici and A. Pfeffer. Simultaneously modeling humans' preferences and their beliefs about other's preferences. *AAMAS*, 2008.

[4] Y. Gal and A. Pfeffer. Predicting people's bidding behavior in negotiation. *AAMAS*, 2006.

[5] Y. Gal and A. Pfeffer. Modeling reciprocity in human bilateral negotiation. *AAAI-07*, 2007.

[6] Y. Gal, A. Pfeffer, F. Marzo, and B. Grosz. Learning social preferences in games. *AAAI*, 2004.

[7] B. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. *AAMAS*, 2004.

[8] W. Guth, R. Schmittberger, and B. Schwarz. An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior and Organization*, (3):367–388, 1982.

[9] J. Kagel and A. Roth. *The handbook of experimental economics*. Princeton Univ. Press, 1995.

[10] L. Katz, editor. *Evolutionary Origins of Morality: Cross-Disciplinary Perspectives*. Impr. Ac., 2000.

[11] R. Katz and S. Kraus. How automated agents treat humans and other automated agents: An experimental study. *AAMAS*, 2008.

[12] A. Sanfey, J. Rilling, J. Aronson, L. Nystrom, and J. Cohen. The neural basis of economic decision-making in the ultimatum game. *Science*, (300):1755–1758, 2003.

[13] E. Sober and D. Wilson. *Unto Others: The Evolution and Psychology of Unselfish Behavior*. Harvard Univ. Press, 1998.

Extended Abstracts

BNAIC 2008

An Architecture for Peer-to-peer Reasoning

George Anadiotis Spyros Kotoulas Ronny Siebes

Department of Artificial Intelligence, Vrije Universiteit Amsterdam, The Netherlands

Abstract

In this extended abstract [1], we focus on achieving scalable reasoning for the Semantic Web through distribution. Most existing approaches split ontologies into triples and distribute them among peers participating in a structured peer-to-peer overlay. Identifying a series of drawbacks with this, we propose an alternative model where each peer maintains control of its ontologies.

1 Introduction

Recently, a series of assumptions when dealing with reasoning on a web scale have been identified [2], which are also largely present in infrastructures developed for the Semantic Web. These are *Small set of axioms*, e.g. limited number of concepts/relationships, *Small number of facts*, e.g. limited number of instance data, *Trustworthiness, correctness, completeness and consistency of facts and axioms*, implying some sort of central control or management and *Static domains*, e.g. infrequent updates or fixed semantics.

Aspiring towards a truly usable and global Semantic Web, research has turned into a number of directions such as approximation, trust infrastructures, database technologies and distributed storage and reasoning. Our focus is on the latter using Peer-to-Peer technology (P2P): we can tap into the vast resources offered by p2p systems to develop scalable infrastructures for the Semantic Web, capitalizing on their advantages of *low cost*, through the distribution of computation and self-organization, *no single point of failure*, due to their symmetric functionality and redundancy, *no single point of administration or control*, making censorship or preferential disclosure of information impossible and, under some conditions, *scalability*, due to the fact that the network can grow on demand.

A number of approaches has already been suggested that also use p2p as a means for distributed reasoning, however they focus on how to efficiently distribute large numbers of triples among peers in the network. We argue against this approach, claiming that although it solves scalability issues concerning the number of facts in the system, it fails to address the rest of the issues mentioned above. Efficient as they may be in storing instance data and ontologies, these approaches do not address scalability in reasoning, do not deal with provenance of information and do not support user/peer control over their own data.

2 Distributed reasoning on the ontology level

Having identified a set of limitations that could inhibit the development of the Semantic Web on current infrastructures, we propose an alternative paradigm, by shifting the level of granularity for peer data from triples to entire ontologies. We propose a model where peers retain control of their ontologies and reasoning is done in a p2p manner. Query resolution is done in an incremental and distributed manner.

All peers have reasoning capabilities and are able to decide when they have had enough answers and query processing should be finished. Furthermore, queries can be decomposed into triple patterns (e.g. `<?, type_of, mtv:music>`). Query resolution consists of 6 steps, which we shall explain by going through a simple example, the resolution of the query

```
SELECT X WHERE X type_of mtv:music
```

[1] In: Proceedings of 1st Workshop on New forms of reasoning for the Semantic Web, International Semantic Web Conference, Busan, Korea, 2007

using RDFS reasoning rules (i.e. this query should return all X that are the predicate of a "type_of" relationship with object being `mtv:music` or any of its subclasses).

1. Partition query and select sub-query Initially, the part of the query to be resolved first needs to be determined. Our example query can be written in a triple pattern format as `<?,type_of, mtv:music>`, that we identify as Θ. Obviously, there is no point in splitting this query further.

2. Determine concepts and relationships required for reasoning Out of the triple pattern Θ, we need to select a starting point for routing our query. Intuitively, the best choice would be `mtv:music`, since it is more selective. Note also that instead of `mtv:music`, we may have a literal and not a concept, in which case we would need to anchor it to a concept.

3. Localize concepts and relationships required for reasoning Either the triples that match the pattern have to be retrieved or the query should be forwarded to the peer(s) that store the ontology(-ies) with these triples. In our example, as in most cases, it is wiser to forward the query, since it is much smaller in size (just a single pattern with no results so far).

4. Perform reasoning locally Now reasoning can be performed locally and the first results can be returned.

5. Determine if answers are adequate The next choice is whether the retrieved results were enough for the user or application. If enough results were found, query resolution stops, otherwise, the query is reformulated to be further processed.

6. Reformulate query To retrieve additional results and according to RDFS semantics, instances that have a type which is a subclass of mtv:music should be returned[2]. Therefore, we can reformulate the query as follows:

```
SELECT X WHERE X type_of Y and Y subclass_of mtv:music
```

Alternatively, the local peer may start a new search for

```
SELECT Y WHERE Y subclass_of mtv:music
```

and continue processing once it gets back the results.

In our system, ontology descriptions or part of ontologies (i.e. concepts or relationships) are stored in a *(*Distributed Hash Table)[3], which can be seen as an efficient (in terms of the number of messages) distributed storage and retrieval mechanism. This index is used to *resolve URIs to locations*, i.e. locating the peers containing the ontologies and instance data for each relationship, concept or instance and to *Anchor terms to concepts in ontologies*, in case we want to anchor literals to ontology concepts or relationships.

Each peer stores a number of ontologies. Although ontologies may be moved across peers and replicated, this is not necessary - i.e. peers may choose to retain complete control over their ontologies or replicate them for performance. By analyzing statistics published by Swoogle [1] about the ontologies it stores, we confirm our hypotheses that ontologies are not strongly connected, which means that in most cases the possible answers can be found locally on a single peer hosting the ontology/-ies of interest, and also that in most cases local reasoning only needs to be done over a relatively small number of ontologies.

We have presented a new method for distributed ontology storage and querying which has ontologies as the normal level of granularity for data distribution. Examining the ontologies currently on the Internet indicates that local reasoning is, most of the times, sufficient for query resolution. Emulations of our approach show that it clearly outperforms (in terms of the number of messages needed to retrieve a comparable set of results) those that rely on storing the full triple distribution on top of DHT. As future work, we intend to evaluate our approach by doing simulation and emulation experiments, introduce some optimizations to the system (triple/ontology caches, semantic topology), as well as examine the scenario where peers do not have the capacity to store their own ontologies/instance data.

References

[1] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM 04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652659, New York, NY, USA, 2004. ACM Press.

[2] Dieter Fensel and Frank van Harmelen. Unifying reasoning and search to web scale. *IEEE Internet Computing*, 11(2):9496, March/April 2007.

[3] Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials*, IEEE, pages 7293, 2004.

[2]Note that this is not the only RDFS rule that applies in this case; for instance, we could look for subclasses of the relationship

Enhancing the Performance of Maximum–Likelihood Gaussian EDAs Using Anticipated Mean Shift

Peter A.N. Bosman[a] Jörn Grahl[b] Dirk Thierens[c]

[a] *Centre for Mathematics and Computer Science, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands,* Peter.Bosman@cwi.nl
[b] *University of Mannheim, Mannheim, Germany,* joern.grahl@bwl.uni-mannheim.de
[c] *the Institute of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands,* Dirk.Thierens@cs.uu.nl

The full version of this paper has been accepted for publication at the *2008 Parallel Problem Solving From Nature — PPSN X* conference.

1 Introduction

Estimation–of–Distribution Algorithms (EDAs) are a specific type of Evolutionary Algorithm (EA). EDAs are characterized by the way in which new solutions are generated. The information in all selected solutions is combined at once. To this end, an interim representation that compresses and summarizes this information is used: a probability distribution over the solution space. New solutions are generated by sampling.

Efficient optimization is guaranteed under suitable conditions [2]. In practice it is however impossible to meet these conditions in general because arbitrarily complex distributions are required. Hence, practical techniques are required. In this paper, we focus on optimization of numerical functions using continuous distributions. The use of the normal distribution or combinations thereof is the most commonly adopted choice. It has already been so since the first EDAs in continuous spaces were introduced. An important question is how efficient EDAs are in the continuous domain using such practical distributions.

2 Maximum–likelihood estimate and descent direction mismatch

Most EDAs have been benchmarked using initialization ranges (IRs) centered around the optimum. An EDA based on the normal distribution with maximum–likelihood (ML) estimates focuses its search by contracting the region of exploration towards its estimated mean. Hence, problems and the search bias of the EDA are very favorably matched, leading to possibly overenthousiastic conclusions.

A simple opposite of a symmetric function is the linear slope. Consider two dimensions, i.e. $f(\boldsymbol{x}) = x_0 + x_1$. The direction $\boldsymbol{u}$ of steepest descent obeys $u_0 = u_1$ and $u_i \leq 0$. Thus, it is most efficient to have the density ellipsoids parallel to and elongated along the line $x_0 = x_1$.

Figure 1 shows the density contours in the case of the full covariance matrix for the first six subsequent generations. The density contours shown are the 95% error ellipses. When ML estimates are used only, the normal distribution quickly contracts. It was previously found that an important reason for this is the premature shrinking of the variance at an exponential rate. Remedies were subsequently formulated [1] (i.e. Adaptive Variance Scaling (AVS) and Standard–Deviation Ratio triggering (SDR)). These remedies enlarge the covariance matrix beyond the ML estimate to prevent premature convergence. Initially, the population is spread uniformly in a square. On a two–dimensional slope the selected solutions form a triangle. Fitting a normal distribution with ML results in density contours aligned in the worst way. Scaling the covariance matrix almost solely increases search effort in the futile direction perpendicular to the best direction.

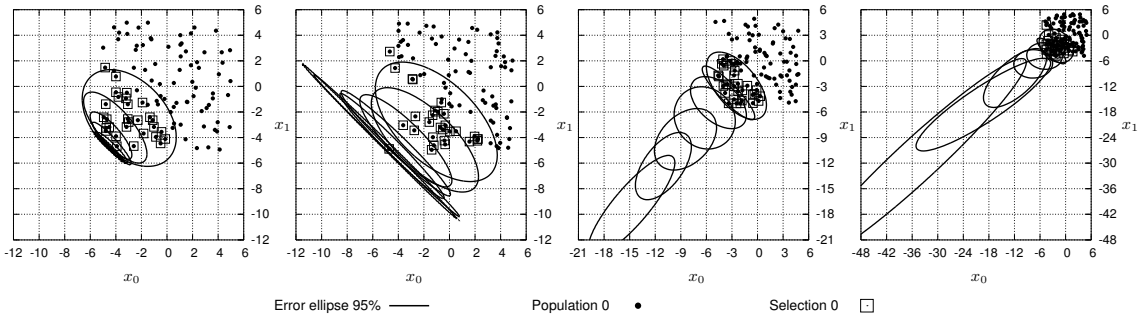


Figure 1: Estimated normal distribution in the first 6 generations of typical runs with (from left to right): ML estimates, SDR–AVS, AMS and AMaLGaM on the two–dimensional slope $f(\boldsymbol{x}) = x_0 + x_1$. The density contours are the 95% error ellipses. Also shown are the population and selection in generation 0.

3 Anticipated Mean Shift and AMaLGaM

On a slope it makes sense to accelerate the search. The AVS scheme provides a principled way to achieve this. A multiplier $c^{\text{Multiplier}}$ is maintained and $c^{\text{Multiplier}}\hat{\boldsymbol{\Sigma}}$ is used instead of $\hat{\boldsymbol{\Sigma}}$ upon sampling. If improvements occur far away from the mean in subsequent generations, $c^{\text{Multiplier}}$ is enlarged. This relation with improvements allows $c^{\text{Multiplier}}$ to be seen as a general accelerator. To accelerate the search on a slope in the right direction, solutions $\boldsymbol{x}$ are altered after sampling using

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + c^{\text{Multiplier}}\delta\hat{\boldsymbol{\mu}}^{\text{Shift}}(t) \tag{1}$$

where $\hat{\boldsymbol{\mu}}^{\text{Shift}}(t) = \hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t-1)$ is a straightforward anticipation of the mean shift and $\delta = 2$ (for the motivation behind this setting, see the full version of this paper). Without $c^{\text{Multiplier}}$ in Equation 1 we call this technique Anticipated Mean Shift (AMS). In Figure 1 the effect of combining AVS with AMS can be seen when traversing the slope in two dimensions. The distribution gets rotated and elongated along the direction of improvement much faster than without the use of the distribution multiplier (note the difference in scale on both axes). Premature convergence due to inefficient sampling that results from fitting only the set of selected solutions without considering the direction of descent is prevented. We name this composite AMS–SDR–AVS technique AMaLGaM (Adapted Maximum–Likelihood Gaussian Model).

We conducted an experimental scalability analysis. On a benchmark–suite of 10 problems we determined the population size required to find the optimum in as little evaluations as possible. Not only is AMaLGaM more efficient than earlier EDAs, it is also robust to rotations and translations of the search space, a property earlier EDAs did not have because of the issues pointed out in this paper. Furthermore, the algorithm is found to be competitive with the current state–of–the–art algorithms in EA research. Detailed results are given in the full version of this paper.

4 Conclusions

Using maximum–likelihood (ML) estimates for the normal distribution in an EDA, premature convergence is likely to occur. Optimization is only performed properly if the initialization range brackets the optimum. Optimization then mainly proceeds by contraction. Methods of adaptive variance scaling (AVS) provide a way to control the rate of contraction and turn it into expansion. Because ML estimates shape the density similar to the configuration of the selected solutions, the density contours can however be misaligned with the direction of descent. The variance then needs to be scaled to excessively large values to still make progress. We have proposed a simple, yet effective approach called anticipated mean shift (AMS) that removes this inefficiency. AMS advances sampled solutions in the direction of the mean shift of the previous generation. We analyzed this technique and provided rational settings for its parameters. We called the resulting EDA Adapted Maximum–Likelihood Gaussian Model — Iterated Density–Estimation Evolutionary Algorithm (AMaLGaM–ID$\mathbb{E}$A or AMaLGaM for short). AMaLGaM makes an important step in the progression of continuous EDAs for numerical optimization.

References

[1] P.A.N. Bosman, J. Grahl, and F. Rothlauf. SDR: A better trigger for adaptive variance scaling in normal EDAs. In D. Thierens et al., editors, *Proc. o/t Genetic and Evol. Comp. Conf. (GECCO)*, pages 492–499. ACM Press, 2007.

[2] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

Modeling the Dynamics of Mood and Depression (extended abstract)[1]

Fiemke Both Mark Hoogendoor Michel Klein Jan Treur

Vrije Universiteit Amsterdam
Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{fboth,mhoogen,mcaklein,treur}@cs.vu.nl

Traditionally, emotions were often left out of consideration in the areas of cognitive, agent, and user modeling. Only few computational models of mood and depression have been developed [2]. However, emotions undoubtedly influence the behavior of humans. In recent years, there is a growing awareness that emotions play a role as well.

Modeling of emotions is important for the development of agents that should exhibit human-like behavior. For example, agents that are used in Intelligent Tutoring Systems to train humans should behave human-like, and show emotions as well. Similarly, virtual agents in games that should interact in a realistic manner have to incorporate the effect of emotions on behavior.

Secondly, also systems that reason over the state of humans should take the emotions and mood of humans into account. For example, one may think of ambient intelligent applications that react on the mood and emotional state of humans. Another category of applications that use knowledge about the mood of humans are systems that support therapy, such as systems the help to quit smoking addiction, or internet-based therapy for depression (online counseling systems).

In this paper, a formal model is presented that can be used to simulate the dynamics in the mood of humans, and more specifically, whether they develop longer periods a undesired moods, as in depressions. To come to this model, commonly used psychological theories about uni-polar (i.e., uncomplicated) depressions are studied. The main concepts and relations are extracted from the theories and represented in a formal model of the aspects of mood and depression.

In the model, depicted in Figure 1, it is assumed that every situation has an emotional value, which represents the extent to which a situation is experienced as something positive. The *objective emotional value of situation* (OEVS) represents how an average person would perceive the situation. How one perceives the situation (*subjective emotional value of situation,* SEVS) influences the *mood* one is in and the *thoughts* one has. By changing or choosing a situation, one can influence their own mood level (e.g. choosing to go to a birthday party when one feels down increases the mood level). This regulation system is based on a current *mood level*, a *prospected mood level* (split into long and short term) and a *sensitivity* representing the ability to choose optimal situations.

For this model there are two decay factors: *diatheses* for downward regulation and *coping* for upward regulation. The term *diatheses* represents the vulnerability one has for developing a depression. The term *coping* represents the skills one has to deal with negative moods and situations. For more details on the mood and depression model, see [1].

The model has been used to simulate different scenarios with different personal characteristics. The different personality settings result in different effects of stressful events on the (long-term) mood of a person.

A mathematical analysis has been performed by rewriting the model in a continuous form of the system of (nonlinear) differential equations. The analysis showed the existence of different equilibriums in the model for persons with different characteristics. A number of properties derived from the most important psychological theories have been formally described using the Temporal Trace Language (TTL, [3]). By

[1] The full version of this paper appeared in [1].

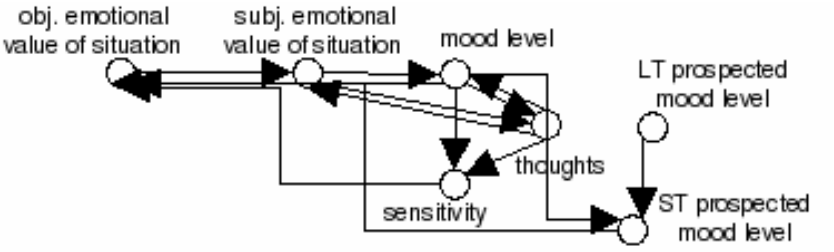


Figure 1: Model of mood dynamics

verifying these properties against a number of representative simulation traces, the adherence of the model to the most important ideas in the theories was validated. For more details on the simulations, the mathematical analysis and the formal properties, see [1].

The resulting model can be useful for developing human-like virtual agents. In addition, models like these can be used in systems that use knowledge about the mood of humans, such as systems for internet-based therapy. Currently, only the major general theories about depression have been involved. For specific applications, it might be required to use more detailed theories that focus on certain aspects. In future work will be investigated whether a model in line with the presented model can help to improve an existing online counselling system.

References

[1] Both, F., Hoogendoorn, M., Klein, M.A., and Treur, J., Formalizing. *Dynamics of Mood and Depression. In: Ghallab, M. (ed.) Proceedings of the 18th European Conference on Artificial Intelligence, ECAI´08*. IOS Press, 2008.

[2] Davidson, R.J., D.A. Lewis, L.B. Alloy, D.G. Amaral, G. Bush, J.D. Cohen, W.C. Drevets, M.J. Farah, J. Kagan, J.L. McClelland, S. Nolen-Hoeksema & B.S. Peterson (2002). *Neural and behavioral substrates of mood and mood regulation. Biological Psychiatry*, *Volume 52, Issue 6*, pp. 478-502.

[3] Bosse, T., C.M. Jonker, L. van der Meij, A. Sharpanskykh and J. Treur. *Specification and Verification of Dynamics in Cognitive Agent Models* In: *Proc. of IAT'06*, IEEE C.Soc. Press, pp.247-254, 2006. Extended version to appear in International Journal of Cooperative Information Systems. In press, 2008.

A Tractable Hybrid DDN-POMDP approach to Affective Dialogue Modeling for Probabilistic Frame-based Dialogue Systems

Trung H. Bui Mannes Poel Anton Nijholt Job Zwiers

University of Twente, P.O.Box 217 7500AE Enschede

Abstract

Designing and developing affective dialogue systems have been receiving much interest from the dialogue research community [1]. Previous work was mainly focused on showing the *system*'s emotion to the user in order to achieve designer's goals such as helping students practice nursing tasks [7] or persuading users to change their dietary behavior [6]. A more challenging issue is to develop a robust dialogue manager that can infer the *user*'s affective state and adapt the system behavior accordingly. For example, in the automatic call routing domain, if the system detects that the user is irritated, the system should verify the information provided by the user or forward the call to a human operator [2].

Recent work from the literature [5, 12] has demonstrated that Partially Observable Markov Decision Processes (POMDPs) are suitable for use in designing this type of dialogue manager for three main reasons. First, the POMDP model allows for realistic modeling of the user's affective state, the user's intention, and other user's hidden state components by incorporating them into the state space. Second, the POMDP-based dialogue manager is able to cope well with *uncertainty* that can occur at many levels inside a dialogue system from the automatic speech recognition and natural language understanding to the dialogue management. Third, the POMDP environment can be used to create a *simulated* user which is useful for learning and evaluation of competing dialogue strategies [9].

However, solving the POMDP problem (i.e. finding a near-optimal policy) for realistic dialogue systems is computationally expensive [4]. Therefore, almost all developed POMDP-based dialogue management approaches (mainly for spoken dialogue systems) are limited to frame-based, and almost toy like, dialogue problems with the size of only a few slots (e.g., two slots in [10], three slots in [13], and four slots in [8]). Recent work [11] has tried to solve this problem by compressing the POMDP structure. However, the affective dialogue model requires a more complex POMDP structure than that of the spoken counterpart. Compressing the POMDP structure prevents us to incorporate a rich model of the user's affect into the state space and might loose dependencies between the user's emotion, goal, and other hidden state variables.

In this paper [3], we propose a novel approach to developing a tractable affective dialogue management model for probabilistic frame-based dialogue systems without compressing the POMDP structure. The model, based on POMDP and Dynamic Decision Network (DDN) techniques, is composed of two main parts: the slot level dialogue manager and the global dialogue manager. The first part is composed of a set of slots where each slot is first modeled as a POMDP and then approximated by a set of DDNs. The second part is handcrafted. The model has two new features: (1) being able to deal with a large number of slots and (2) being able to take into account some aspects of the user's affective state in deriving the adaptive dialogue strategies.

Our implemented prototype dialogue manager can handle hundreds of slots in real time (i.e., processing time is smaller than one second), where each individual slot might have hundreds of values. The approach is illustrated through a route navigation example in the crisis management domain. We conducted various experiments to evaluate our approach and to compare it with state-of-the-art approximate POMDP techniques and three handcrafted policies. The results showed that the DDN-POMDP policy outperforms the handcrafted policies when the user's action error is induced by stress as well as when the observation error of the user's action increases. Further, performance of the one-step look-ahead DDN-POMDP policy after tuning its internal reward is close to the approximate POMDP counterparts. The method is not only useful for building affective dialogue systems but also applicable for the development of robust dialogue managers for multimodal dialogue systems, in particular spoken dialogue systems.

References

[1] Elisabeth André, Laila Dybkjær, Wolfgang Minker, and Paul Heisterkamp, editors. *Affective Dialogue Systems, Tutorial and Research Workshop (ADS 2004), Kloster Irsee, Germany, June 14-16, 2004, Proceedings*, volume 3068 of *Lecture Notes in Computer Science*. Springer, 2004.

[2] Anton Batliner, K. Fischer, Richard Huber, Jörg Spilker, and Elmar Nöth. How to find trouble in communication. *Speech Communication*, 40(1-2):117–143, April 2003.

[3] Trung H. Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable hybrid DDN-POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 2008, to appear.

[4] Trung H. Bui, Boris van Schooten, and Dennis Hofs. Practical dialogue manager development using POMDPs. In S. Keizer, H.C. Bunt, and T. Paek, editors, *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue (SIGdial '07)*, pages 215–218, Antwerp, Belgium, 2007.

[5] Trung H. Bui, Job Zwiers, Mannes Poel, and Anton Nijholt. Toward affective dialogue modeling using partially observable Markov decision processes. In D. Reichardt, P. Levi, and J.-J.C. Meyer, editors, *Proceedings of Workshop Emotion and Computing, 29th Annual German Conference on Artificial Intelligence*, pages 47–50, Bremen, Germany, 2006.

[6] Fiorella de Rosis, Nicole Novielli, Valeria Carofiglio, Addolorata Cavalluzzi, and Berardina D. Carolis. User modeling and adaptation in health promotion dialogs with an animated character. *Journal of Biomedical Informatics*, 39(5):514–531, October 2006.

[7] Dirk Heylen, Anton Nijholt, and Rieks op den Akker. Affect in tutoring dialogues. *Applied Artificial Intelligence*, 19:287–311, 2005.

[8] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-00)*, pages 93 – 100, Hong Kong, China, 2000. ACL.

[9] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*, 21(2):97–126, 2006.

[10] Jason Williams and Steve Young. Scaling up POMDPs for dialogue management: the summary POMDP method. In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU '05)*, pages 250–255, Cancún, Mexico, 2005.

[11] Jason D. Williams and Steve Young. Scaling POMDPs for dialog management with composite summary point-based value iteration (CSPBVI). In *Proceedings of the AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, pages 37–42, Boston, MA, USA, 2006. The AAAI Press.

[12] Jason D. Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.

[13] Bo Zhang, Qingsheng Cai, Jianfeng Mao, and Baining Guo. Spoken dialog management as planning and acting under uncertainty. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH '01)*, pages 2169–2172, Aalborg, Denmark, 2001.

An Algorithm for Semi-Stable Semantics

Martin Caminada [a]

[a] *University of Luxembourg*

Formal argumentation, in its most abstract form, is done using a set of abstract arguments and a defeat relation between these arguments. Since an argument A may be defeated by another argument B which may in its turn be defeated by a third argument C, the status of A (whether it can be accepted or not) partly depends on the status of C. Thus, what is needed is an overall criterion for determining which of the arguments can be considered to be ultimately justified. Several of such criteria have been proposed, the most well-known of these are grounded, preferred and stable semantics [1]. A relatively new proposal is semi-stable semantics [2]. Semi-stable semantics can be placed between stable semantics and preferred semantics, as every stable extension is also a semi-stable extension and every semi-stable extension is also a preferred extension. Moreover, if there exists at least one stable extension, then the semi-stable extensions will be exactly the same as the stable extensions. The advantage of semi-stable semantics, however, is that for finite argumentation frameworks semi-stable extensions always exist, even in situations where stable extensions do not exist.

Definition 1. *An* argumentation framework *is a pair* (Ar, def) *where* Ar *is a finite set of arguments and* $def \subseteq Ar \times Ar$. *We say that an argument* A defeats *an argument* B *iff* $(A, B) \in def$.

Definition 2 (defense / conflict-free)**.** *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ *is* conflict-free *iff* $\neg\exists A, B \in \mathcal{A}rgs : A$ *defeats* B. $\mathcal{A}rgs$ defends $A \in Ar$ *iff* $\forall B \in Ar : (B$ *defeats* $A \supset \exists C \in \mathcal{A}rgs : C$ *defeats* $B)$.

Definition 3. *Let* (Ar, def) *be an argumentation framework.* $\mathcal{A}rgs \subseteq Ar$ *is*
- an admissible set *iff* $\mathcal{A}rgs$ *is conflict free and defends each* $A \in \mathcal{A}rgs$*,*
- a preferred *extension iff* $\mathcal{A}rgs$ *a maximal admissible set (w.r.t set inclusion),*
- a stable *extension iff* $\mathcal{A}rgs$ *is conflict-free and defeats each argument in* $Ar \backslash \mathcal{A}rgs$*, and*
- a semi-stable *extension iff* $\mathcal{A}rgs$ *is an admissible set where* $\mathcal{A}rgs \cup \{B \mid \exists A \in \mathcal{A}rgs : A$ *defeats* $B\}$ *is maximal (w.r.t. set-inclusion).*

Definition 4. *A labelling is a total function* $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. *An argument is illegally* `in` *iff it is labelled* `in` *but not all its defeaters are labelled* `out`. *An argument is illegally* `out` *iff it is labelled* `out` *but it does not have a defeater that is labelled* `in`*. A labelling is called* admissible *iff no argument is illegally* `in` *or illegally* `out`*. A labelling is called* preferred *iff it is admissible and the set of* `in`*-labelled arguments is maximal (w.r.t. set inclusion). A labelling is called* stable *iff it is admissible and does not contain an argument labelled* `undec`*. A labelling is called* semi-stable *iff it is admissible and the set of* `undec`*-labelled arguments is minimal (w.r.t. set-inclusion). An argument is called superillegally* `in` *iff it is labelled* `in` *and is defeated by an argument that is labelled* `in` *but is not illegally* `in`*.*

It can be shown that admissible labellings correspond with admissible sets, preferred labellings correspond with preferred extensions, stable labellings correspond with stable extensions and semi-stable labellings correspond with semi-stable extensions.

Now that the preliminary concepts have been explained, it is time to treat the main question of how to compute, given an argumentation framework, all its semi-stable labellings. The idea is to do this by generating a set $\mathcal{L}abellings$ of admissible labellings that includes at least all semi-stable labellings. We then have to select those labellings in $\mathcal{L}abellings$ with minimal `undec` to obtain the final answer.

How does one generate an admissible labelling? A possible approach is to start with the all-in labelling (the labelling in which every argument is labelled `in`). This labelling trivially satisfies the absence of arguments that are illegally `out`. However, for an admissible labelling also the absence of arguments that

are illegally in is required, and the all-in labelling may contain many arguments that are illegally in. This means we need a way of changing the label of an argument that is illegally in, preferrably without creating any arguments that are illegally out. This is done using a sequence of *transition steps*. A transition step basically takes an argument that is illegally in and relabels it to out. It then checks if, as a result of this, one or more arguments have become illegally out. If this is the case, then these arguments are relabelled to undec. Thus, each transition step preserves the absence of arguments that are illegally out.

Definition 5. *Let $\mathcal{L}$ be a labelling and A an argument that is illegally* in *in $\mathcal{L}$. A* transition step *on A in $\mathcal{L}$ consists of the following: (1) the label of A is changed from* in *to* out*, and (2) for every $B \in \{A\} \cup \{C \mid A$ defeats $C\}$, if B is illegally* out*, then change the label of B from* out *to* undec*.*

Since the algoritm starts with the labelling in which every argument is labelled in, we assume the presence of the constant all_in, which stands for the all-in labelling. The global variable pot_semi-stables stands the potential semi-stable labellings, that is, the admissible labellings with minimal undec that have been found until now. If, during the search algorithm, one finds that the current labelling is worse (that is: it has a proper superset of undec labelled arguments) than an admissible labelling found earlier, then it is time to stop evaluating the current transition sequence, since its final result will not be semi-stable anyway.

If there is no argument that is illegally in then we are at the end of a terminated transition sequence and have obtained an admissible labelling. From the previous check, we already know that this admissible labelling is not any worse than what we already have found (it does not have a proper superset of undec labelled arguments compared to a previously computed admissible labelling), so we add it to the set of potential semi-stable labellings (pot_semi-stables). We then have to check if we found something that is actually *better* than what we found earlier. If so, we need to delete some of the old results (remove it from pot_semi-stables).

```
01.  pot_semi-stables = ∅; find_semi-stables(all-in);
02.  print pot_semi-stables; end;
03.
04.  procedure find_semi-stables(L)
05.     # if we have something worse than found earlier,
06.     # then prune the search tree and backtrack
07.     if ∃L' ∈ pot_semi-stables:  undec(L') ⊊ undec(L) then return;
08.     # now see if the transition sequence has terminated
09.     if L does not have an argument that is illegally in then
10.       for each L' ∈ pot_semi-stables
11.         # if old result is worse than new labelling:  remove
12.         if undec(L) ⊊ undec(L') then
13.           pot_semi-stables := pot_semi-stables - L'; endif;
14.       endfor;
15.       # add our newly found labelling as a candidate; we already
16.       # know that it is not worse than what we already have
17.       pot_semi-stables := pot_semi-stables ∪ L;
18.       return; # we are done with this one; try next possibility
19.     else
20.       if L has an argument that is superillegally in then
21.         A := some argument that is superillegally in in L;
22.         find_semi-stables(transition_step(A, L));
23.       else
24.         for each argument A that is illegally in in L
25.           find_semi-stables(transition_step(A, L)); endfor;
26.  endif; endif; endproc;
```

References

[1] Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77** (1995) 321–357

[2] Caminada, M.: Semi-stable semantics. In Dunne, P., Bench-Capon, T., eds.: Computational Models of Argument; Proceedings of COMMA 2006, IOS Press (2006) 121–130

Towards an Argument Game for Stable Semantics

Martin Caminada [a], Yining Wu [a]

[a] *University of Luxembourg*

Formal argumentation, in its most abstract form, is done using a set of abstract arguments and a defeat relation between these arguments. Since an argument A may be defeated by another argument B which may in its turn be defeated by a third argument C, the status of A (whether it can be accepted or not) partly depends on the status of C. Thus, what is needed is an overall criterion for determining which of the arguments can be considered to be ultimately justified. One of the oldest approaches for this is *stable semantics*, a concept that goes back to [3]. In the current paper we present a discussion game that can be used to determine whether an argument is in at least one stable extension. Our approach is based on [4], where a similar discussion game for preferred semantics is specified. Although the approach of [4] could also be used directly for the purpose of stable semantics, this only works in argumentation frameworks where each preferred extension is also a stable extension. Our current approach does not have this restriction.

Definition 1. *An* argumentation framework *is a pair* (Ar, def) *where* Ar *is a finite set of arguments and* $def \subseteq Ar \times Ar$. *We say that an argument* A defeats *an argument* B *iff* $(A, B) \in def$.

Definition 2. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ *is* conflict-free *iff* $\neg\exists A, B \in \mathcal{A}rgs : A$ *defeats* B. $\mathcal{A}rgs$ defends $A \in Ar$ *iff* $\forall B \in Ar : (B$ *defeats* $A \supset \exists C \in \mathcal{A}rgs : C$ *defeats* $B)$. $\mathcal{A}rgs$ *is an* admissible set *iff* $\mathcal{A}rgs$ *is conflict free and defends each* $A \in \mathcal{A}rgs$. $\mathcal{A}rgs$ *is a* stable extension *iff* $\mathcal{A}rgs$ *is an admissible set and defeats each argument in* $Ar \backslash \mathcal{A}rgs$.

Definition 3. *A* labelling *is a total function* $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. *An* admissible labelling *is a labelling where each argument that is labelled* `in` *has all its defeater labelled* `out`*, and each argument that is labelled* `out` *has at least one defeater that is labelled* `in`. *A* stable labelling *is an admissible labelling without any argument labelled* `undec`.

It can be shown that admissible labellings correspond with admissible sets and stable labellings correspond with stable extensions [1, 2].

Theorem 1. *Let* (Ar, def) *be an argumentation framework and let* $\mathcal{A}rgs \subseteq Ar$.
$\mathcal{A}rgs$ *is an admissible set iff there exists an admissible labelling* $\mathcal{L}$ *with* $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$.
$\mathcal{A}rgs$ *is a stable extension iff there exists a stable labelling* $\mathcal{L}$ *with* $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$.

Now we present a discussion game for argumentation under stable semantics. Our work is inspired by Vreeswijk and Prakken, who have defined a similar game for preferred semantics [4].

Definition 4. *Let* (Ar, def) *be an argumentation framework. A* stable discussion *is a sequence of moves* $[M_1, M_2, \ldots, M_n]$ *(*$n \geq 0$*) such that:*

- *each* M_i *(*$1 \leq i \leq n$*) where* i *is odd (which is called a* proponent move*) is of the form* $\texttt{in}(A)$*, where* $A \in Ar$.
- *each* M_i *(*$1 \leq i \leq n$*) where* i *is even (which is called an* opponent move*) is of the form* $\texttt{out}(A)$ *where* $A \in Ar$*, or of the form* $\texttt{question}(A)$ *where* $A \in Ar$.
- *For each opponent move* $M_i = \texttt{out}(A)$ *(*$2 \leq i \leq n$*) there exists a proponent move* $M_j = \texttt{in}(B)$ *(*$j < i$*) where* A *defeats* B.

- *For each proponent move* $M_i = \texttt{in}(A)$ *(*$3 \leq i \leq n$*) it either holds that (1)* $M_{i-1} = \texttt{out}(B)$ *where* A *defeats* B*, or (2)* $M_{i-1} = \texttt{question}(B)$ *where either* $A = B$ *or* A *defeats* B*.*
- *For each opponent move* $M_i = \texttt{out}(A)$ *(*$2 \leq i \leq n$*) there does not exist an opponent move* $M_j = \texttt{out}(A)$ *with* $j < i$*.*
- *For each opponent move* $M_i = \texttt{question}(A)$ *(*$2 \leq i \leq n$*) there does not exist any move* M_j *(*$j < i$*) of the form* $\texttt{in}(A)$*,* $\texttt{out}(A)$ *or* $\texttt{question}(A)$*.*
- *For each proponent move* $M_i = \texttt{in}(A)$ *(*$3 \leq i \leq n$*) there does not exist an opponent move* $M_j = \texttt{out}(A)$ *with* $j < i$*.*

A stable discussion $[M_1, M_2, \ldots, M_n]$ *is said to be* finished *iff there exists no* M_{n+1} *such that* $[M_1, M_2, \ldots, M_n, M_{n+1}]$ *is a stable discussion, or if* M_n *is an opponent move of the form* $\texttt{out}(A)$ *for which there exists a proponent move* M_i *(*$1 \leq i < n$*) of the form* $\texttt{in}(A)$*. A finished discussion is won by the proponent if the last move is a proponent move, and is won by the opponent if the last move is an opponent move.*

Theorem 2. *Let* (Ar, def) *be an argumentation framework and* $A \in Ar$*. There exists a stable labelling* $\mathcal{L}$ *with* $\mathcal{L}(A) = \texttt{in}$ *iff there exists a stable admissible discussion for* A *that is won by the proponent.*

Although the above described discussion game is aimed for the credulous variant of stable semantics ("is there *at least one* stable extension which contains argument A?"), it is also possible to apply the discussion game in the context of the sceptical variant of stable semantics ("is argument A contained in *each* stable extension?"). The idea is that an argument is in each stable extension iff there is no stable extension that contains one of its defeaters.
So in order to examine whether an argument A is in each stable extension, one should examine the defeaters of A one by one. If one finds a defeater that is in a stable extension, then the question of whether A is in each extension can be answered with "no". If, however, it turns out that each defeater of A is not in any stable extension, then the answer is "yes". Therefore, one can simply apply the (credulous) stable discussion game for each defeater of A, to obtain the answer regarding sceptical stable.

References

[1] M.W.A. Caminada. On the issue of reinstatement in argumentation. In M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, pages 111–123. Springer, 2006. LNAI 4160.

[2] M.W.A. Caminada. An algorithm for computing semi-stable semantics. In *Proceedings of the 9th European Conference on Symbolic and Quantitalive Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, number 4724 in Springer Lecture Notes in AI, pages 222–234, Berlin, 2007. Springer Verlag.

[3] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[4] G. A. W. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA-00)*, number 1919 in Springer Lecture Notes in AI, pages 239–253, Berlin, 2000. Springer Verlag.

Temporal Extrapolation within a Static Clustering[1]

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

Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands

1 Introduction and background

Predicting the behaviour of individuals is a core business of policy makers. This paper discusses a new way of predicting the "movement in time" of items through pre-defined classes by analysing their changing placement within a static, preconstructed 2-dimensional clustering. It employs the visualization realized in previous steps within item analysis, rather than performing complex calculations on each attribute of each item. For this purpose we adopt a range of well-known mathematical extrapolation methods that we adapt to fit our need for 2-dimensional extrapolation.

It is common practice to visualize analysis results as a clustering within the 2-dimensional plane, approximating the correct, multi-dimensional distances. As long as the error made within the visualization is as small as possible there is no preferred method of obtaining such a clustering within our prediction system. However, the algorithm must allow for incremental addition of single elements from the sequence to be extrapolated, which is not supported by all of the known methods.

All interpolation schemes are suitable starting points for the process of extrapolation. In most cases, it is sufficient to simply continue the fabricated interpolation function after the last existing data point. In the case of a spline interpolation, however, a choice can be made to continue the polynomial constructed for the last interval (which can lead to strange artifacts), or extrapolate with a straight line, constructed with the last known derivative of that polynomial (Figure 1 and 2). In our approach both x and y are coordinates and therefore inherently independent variables. They depend on the current visualization alone. Within our model, they do however depend on the time variable t. Because our methods aim to extrapolate x, y out of one other variable t, we need a form of 2-dimensional extrapolation.

2 Approach

The data used as reference within our approach is represented by a square $q \times q$ distance matrix describing the proximity between all q items. These items are clustered and visualized in a 2-dimensional plane with dots representing our reference items. This step in the approach is done only once so the focus should be on the quality of the clustering instead of the computational complexity. From this point on this clustering is considered to be fixed or static.

Analysis of the behaviour of new items should start with the calculation of the attributes for each time-unit. These units are supposed to be cumulative, meaning that they contain all the item's *baggage*, i.e., its whole history, up to the specified moment. Using the same distance measure that was used to create the initial distance matrix, the *distance vector per time-unit* can now be calculated. This should be done for all t time-units, resulting in t vectors of size q. These vectors can now be visualized as before.

After selecting the best extrapolation scheme for our type of data our method creates a function that extrapolates item behaviour. One advantage of this approach is that the extrapolation or prediction is immediately visualized to the end-user rather than presenting him or her with a large amount of numerical data.

[1] Published in: Foundations of Intelligent Systems, Proceedings of ISMIS 2008, LNAI 4994, pp.189–195, Springer 2008.
This research is part of the DALE (Data Assistance for Law Enforcement) project as financed in the ToKeN program from the Netherlands Organization for Scientific Research (NWO) under grant number 634.000.430.

Figure 1: Straight line extrapolation (left) and polynomial continuation (right)

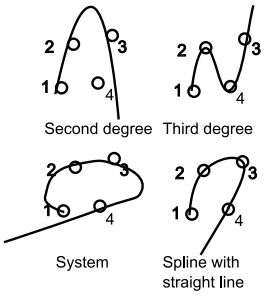


Figure 2: Different extrapolation methods yield very different results

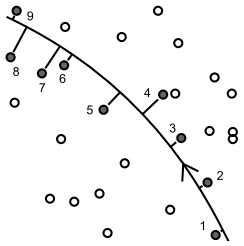


Figure 3: Selecting points with the shortest distance to the extrapolation line

If the user is familiar with the data under consideration, he/she can analyse the prediction in an eye blink. Augmenting the system with a click and point interface would enable the end-user to use the prediction as a starting point for further research. In most cases it is desirable to predict which class the item under consideration might belong to in the future. In that case it is important to retrieve further information from some of the reference items and assign future attribute values and a future class to the item.

A first step would be to select r reference items closest to the extrapolation line. This can easily be done by evaluating the geometric distance of all reference points to the line and selecting those with the smallest distance, see Figure 3. We order these points by their respective distance to the last known data point of the extrapolated item: the confidence of the prediction declines with this distance and calculate the expected future attributes of the item under consideration based upon this weighted average. The extrapolated item can now be visualized into the clustering according to its future attributes and be classified accordingly.

3 Experiments

The detection, analysis, progression and prediction of criminal careers is an important part of automated law enforcement analysis [1]. Our approach of temporal extrapolation was tested on the national criminal record database of The Netherlands. This database contains approximately one million offenders and their respective crimes (approximately 50 types). For each item (i.e., person) in the test set we only consider the first $t = 4$ time periods. The accuracy is described by the mean similarity between the calculated and the expected values of the attributes.

Although the runtime needed for visual extrapolation is much less than that of regular methods, the accuracy is comparable. For this database the best result is the spline extrapolation with a straight line, having a very short runtime while reaching an accuracy of 88.7%.

4 Conclusion and Future Directions

In this paper we demonstrated the applicability of temporal extrapolation by using the prefabricated visualization of a clustering of reference items. We demonstrated a number of extrapolation techniques and employed them to predict the future development of item behaviour. Our methods were tested within the arena of criminal career analysis, predicting the future of unfolding criminal careers. We showed that our novel approach largely outperforms standard prediction methods in the sense of computational complexity, with a loss in accuracy smaller than 1 percentage point. The visual nature of our method enables the analyst of the data to immediately continue his/her research since the prediction results are easily displayed within a simple graphical interface.

References

[1] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Data mining approaches to criminal career analysis. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 171–177, 2006.

Approximating Pareto Fronts by Maximizing the S-Metric with an SMS-EMOA/Gradient Hybrid

Michael Emmerich [a] André Deutz [a] Nicola Beume [b]

[a] *Leiden University, Leiden Institute for Advanced Computer Science, 2333 CA Leiden, The Netherlands, http://natcomp.liacs.nl/*
[b] *University of Dortmund, Chair of Algorithm Engineering, 44221 Dortmund, Germany, ls11-www.cs.uni-dortmund.de*

In multiobjective optimization a solution has to fulfill several objectives in the best possible way. Without loss of generality we restrict our attention to minimization problems. Formally, we have:

$$\mathbf{f} = (f_1, \ldots, f_m)^T, \quad f_1(\mathbf{x}) \to \min, \ldots, f_m(\mathbf{x}) \to \min, \quad \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d. \tag{1}$$

, where each f_i is a real-valued, continuous function on some open subset containing $\mathcal{X}$. The domain $\mathcal{X}$ is called the *decision space* or *search space* and contains all feasible solutions, and the image, $\mathbf{f}(\mathcal{X}) \subseteq \mathbb{R}^m$, is called the *objective space* and is denoted by $\mathcal{Y}$. In $\mathbb{R}^m$ the following partial order can be defined: $\mathbf{y}^{(1)} \prec \mathbf{y}^{(2)}$ iff $\forall i \in \{1, \cdots m\}, \mathbf{y}_i^{(1)} \leq \mathbf{y}_i^{(2)}$ and $\exists i_0 \in \{1, \cdots, m\}$ such that $\mathbf{y}_{i_0}^{(1)} < \mathbf{y}_{i_0}^{(2)}$. The objective space inherits this partial order. In turn we can define the notion of (weak) dominance on the search space: a point $\mathbf{x} \in \mathcal{X}$ (weakly) dominates a point $\mathbf{x}' \in \mathcal{X}$ iff $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}')$. A point $\mathbf{x} \in \mathcal{X}$ strictly dominates a point $\mathbf{x}' \in \mathcal{X}$ iff $\forall i \in \{0, \cdots, m\}, f_i(\mathbf{x})) < f_i(\mathbf{x}')$. The set of points which are minimal with respect to the partial order $\prec$ of a set $A \subseteq \mathbb{R}^m$ is called the Pareto front(PF) of A which we denote by A_N. The set of minimal points (notation $\mathcal{X}_E$) in $\mathcal{X}$ with respect to the weakly dominance relation in $\mathcal{X}$ is called the *efficient set*.

Among the many developed quality measures for PF approximations, the *S-metric* or *dominated hypervolume* by Zitzler and Thiele [3] stands out and is of utmost importance. It is defined as

$$\mathcal{S}(Y) = \text{Lebesgue}\{\mathbf{y} | \exists \mathbf{y}' \in Y : \mathbf{y}' \prec \mathbf{y} \wedge \mathbf{y} \prec \mathbf{y}^{ref}\}, \tag{2}$$

, where $Y \subseteq \mathcal{Y}$, and $\mathbf{y}^{ref}$ is a reference point. For any $X \subseteq \mathcal{X}$, we define $\mathcal{S}(X) = \mathcal{S}(\mathbf{f}(X))$.

The maximization of the S-metric receives increasingly more attention as a solution principle for approximating PFs by means of a well-distributed non-dominated set. Accordingly, the problem of finding a good approximation of the PF of the original multiobjective optimization problem becomes:

$$\mathcal{S}(X) \to \max, X \subseteq_\mu \mathcal{X} \tag{3}$$

where $X \subseteq_\mu \mathcal{X}$ means that X is a set of at most μ elements from $\mathcal{X}$.

The idea to use the differentiability/one-sided differentiability of the $\mathcal{S}$-metric in the search process can be accomplished by viewing PF (or efficient set) approximation sets of finite size as points in $\mathbb{R}^k$ for some suitably chosen k. This view opens up the possibility to use methods which are based on derivative (or gradient) information such as, for example, variations of the line search algorithm.

In order to compute the derivative or gradient of the $\mathcal{S}$-metric, we can represent a *finite* subset Y of $\mathcal{Y}$ consisting of μ points $\mathbf{p}^{(i)} = (y_1^{(i)}, \ldots, y_d^{(i)})^\top$, where $i = 1, \ldots, \mu$ as an element of $\mathbb{R}^{\mu \cdot m}$ as follows: $\mathbf{p} = (y_1^{(1)}, \ldots, y_m^{(1)}, \ldots, y_1^{(\mu)}, \ldots, y_m^{(\mu)})^\top = (p_1, \ldots, p_{\mu \cdot m})^\top$. With this notation the gradient of the S-metric takes the form $\nabla_{\mathbf{p}} \mathcal{S} = (\frac{\partial \mathcal{S}}{\partial p_1}, \ldots \frac{\partial \mathcal{S}}{\partial p_{\mu \cdot d}})^\top$. Furthermore, we can similarly represent a finite subset X of $\mathcal{X}$ of μ points ($\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\mu)}$) as an element $\mathbf{x}$ of $\mathbb{R}^{\mu \cdot d}$. Under these identifications we get $\mathcal{S}(X) =$

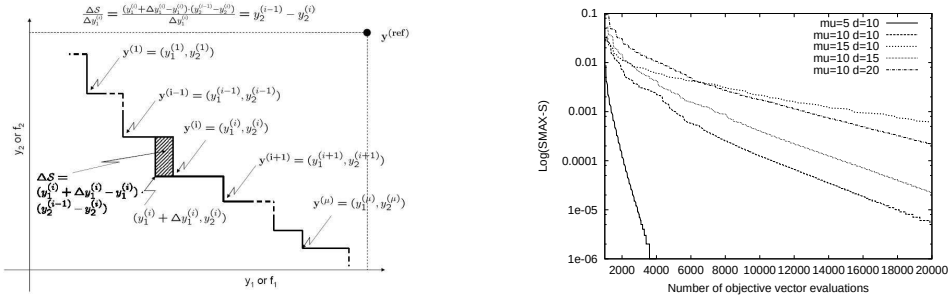


Figure 1: Differentials used in the computation of the gradient in the 2-D objective space (left) and the convergence history of the gradient based search from a starting point generated by the SMS-EMOA for different population sizes and search space dimensions (right).

$\mathcal{S}(\mathbf{F}(X))$, where $\mathbf{F}$ is defined as follows: $\mathbf{F}(\mathbf{x^{(1)}}, \cdots, \mathbf{x}^{(\mu)}) = (\mathbf{f}(\mathbf{x^{(1)}}), \mathbf{f}(\mathbf{x^{(2)}}), \cdots, \mathbf{f}(\mathbf{x}^{(\mu)}))^\top$, and $\mathbf{f} = (f_1, f_2, \cdots, f_m)^\top$ as before. The chain rule for:

$$\mathbb{R}^{\mu \cdot d} \underbrace{\xrightarrow{\mathbf{F}}}_{\text{decision to objective space}} \mathbb{R}^{\mu \cdot m} \underbrace{\xrightarrow{\mathcal{S}}}_{\text{objective space to } S\text{-metric}} \mathbb{R}^{+}. \tag{4}$$

, implies that the gradient of $\mathcal{S}$-metric at $\mathbf{x} \in \mathbb{R}^{(\mu \cdot d)}$ representing a finite subset of $\mathcal{X}$ can be expressed in terms of the gradients of f_i at the sites $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)}$ and the gradient of the $\mathcal{S}$-metric at $\mathbf{F}(\mathbf{x})$.

For the 2D case the (partial) derivatives are readily computed (though we have to be careful with computing the derivatives of the $\mathcal{S}$-metric at approximations sets with weakly but not strictly dominated points: in this case one sided derivatives come to the rescue). In gradient based techniques it is advisable to give influence to dominated points in an PF approximation set. In this case a penalty is introduced in addition to letting the local gradient point to the attainment surface. These measures will make it more attractive for dominated points to move into a non-dominated position.

The gradient-descent method requires a good starting point in order to converge to the Pareto front. For this purpose an EMOA is applied which generates a good approximation of the Pareto front. We propose the SMS-EMOA as it has shown excellent results in the past (cf. [1]). The SMS-EMOA uses a steady-state selection scheme, i.e. in each generation one new solution is generated and one solution is discarded. A population of μ individuals is optimized without additional archives (which are often used in other EMOA). The $\mathcal{S}$-Metric is used within the selection operator to determine the subset of μ individuals with the highest $\mathcal{S}$-Metric value. Thereby, the individual with the least exclusive contribution of dominated hypervolume is discarded. The maximization of the $\mathcal{S}$-Metric results in a well-distributed solution set with an emphasis on solutions in regions with fair trade-offs. The SMS-EMOA's final population functions as the starting point of the gradient strategy which does only a fine-tuning of the solutions.

In [2], the SMS-EMOA and the hybrid are run on the generalized Schaffer problem (with known $\mathcal{S}$-metric optimal set) and on the ZDT benchmark [3]. In the hybrid approach runs are also compared to runs with penalty. First results on test problems show that the problem can approximate PFs with arbitrary precision. Moreover the convergence rate is linear (Figure 1). However, its computation time grows fast with the population size. In conclusion, the new technqiue is promising as a high accuracy technique for Pareto front approximation with a small number of points.

References

[1] N. Beume, B. Naujoks, and M .Emmerich. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *EJOR*, 181(3):1653–1669, 2007.

[2] M. Emmerich, A. Deutz, and N. Beume. Gradient-Based/Evolutionary Relay Hybrid for Computing Pareto Front Approximations Maximizing the S-Metric. In Th. Bartz-Beielstein et al., editor, *Hybrid Metaheuristics*, LNCS 4771, pages 140–156. Springer, Berlin, 2007.

[3] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

A Probabilistic Model for Generating Realistic Lip Movements from Speech

Gwenn Englebienne [a] Tim Cootes [b] Magnus Rattray [b]

[a] *University of Amsterdam*
[b] *The University of Manchester*

Abstract

We model the correspondence between facial motion and speech, where the face and sound are modelled separately and phonemes are the link between both. We propose a sequential model and evaluate its suitability for the generation of the facial animation from a sequence of phonemes, which we obtain from speech. We evaluate the results both by computing the error between generated sequences and real video, as well as with a rigorous double-blind test with human subjects. Experiments show that our model compares favourably to other existing methods and that the generated sequences are comparable to real video sequences.

1 Introduction

Generative systems that model the relationship between face and speech offer a wide range of exciting prospects. Models combining speech and face information have been shown to improve automatic speech recognition [3]. Conversely, generating video-realistic animated faces from speech has immediate applications to the games and movie industries. There is a strong correlation between lip movements and speech, and there have been multiple attempts at generating an animated face to match some given speech realistically [6, 1, 8]. Studies have indicated that speech might be informative not only of lip movement but also of movement in the upper regions of the face [2]. Incorporating speech therefore seems crucial to the generation of true-to-life animated faces.

The subject of this paper [5], is to build a generative probabilistic model, capable of generating realistic facial animations in real time, given speech. We first use an Active Appearance Model (AAM [4]) to extract features from the video frames. The AAM itself is generative and allows us to produce video-realistic frames from the features. We then use a Hidden Markov Model (HMM [7]) to align phoneme labels to the audio stream of video sequences, and use this information to label the corresponding video frames. We propose a model which, when trained on these labelled video frames, is capable of generating new, realistic video from unseen phoneme sequences. Our model is a modification of Switching Linear Dynamical Systems (SLDS [9]) and we show that it performs better at generation than other existing models. We compare its performance to two previously proposed models by comparing the sequences they generate to a golden standard, features from real video sequences, and by asking volunteers to select the "real" video in a forced-choice test.

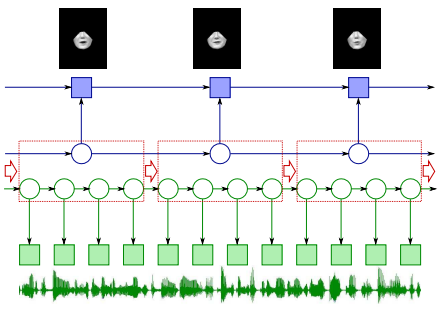


Figure 1: Combining sound and face

The results of human evaluation of our generated sequences are extremely encouraging. Our system performs well with any speech, and since it can easily handle real-time generation of the facial animation, it brings a realistic-looking, talking avatar within reach.

2 Modelling the dynamics of the face

The sequences are split into an audio and a video stream, which are treated separately (see Figure 1). From the sound stream, we extract Mel Frequency Cepstrum Coefficients (MFCC) at a rate of 100Hz. We train a HMM on these MFCC features, and use it to align phonetic labels to the sound. The labels obtained from the sound stream are then used to label the corresponding video frames. The difference in rate (the video is processed at 29.97 frames per second while MFCC are computed at 100 Hz) is handled by simple voting: each video frame is labelled with the phoneme that labels most of the corresponding sound frames.

The feature extraction for the video was done using an Active Appearance Model (AAM [4]). AAMs are statistical models of objects in images which decouple shape and texture, thus allowing a wide range of convincing synthetic faces to be represented and generated [4]. The shape of the lower part of the face is represented by the location of 23 points on key features on the eyes, mouth and jaw-line. In this case a 32 parameter model proves sufficient. Since the AAM parameters are a low-dimensional linear projection of the original object, projecting those parameters back to the high-dimensional space allows us to reconstruct the modelled part of the original image.

We model the face using only phoneme labels to capture the shared information between speech and face. We use 41 distinct phoneme labels, two of which are reserved for audible intake of breath and silence, the rest being the generally accepted phonemes in the English language.

We propose a model which we call the Deterministic Process Dynamical System, which is similar to a switching linear dynamical system but where a deterministic latent process variable is assumed. The resulting model performs much better at generation, in terms of human appreciation, than more flexible systems which incorporate a noise model for the process. We propose efficient techniques for training, and compare the model to well-known probabilistic models of sequential data.

We evaluated the models in two ways: (1) by computing the error between generated face features and a ground truth (the features of real video), and (2) by asking human subjects to rate how they perceived the sequences. Both tests were done on the same real-world data, but the sets are partitioned differently: the comparison to the ground truth was done using 10-fold cross-validation, while the test on humans was done using a single partitioning, due to the limited availability of unbiased test subjects.

3 Conclusion

In this work we have proposed a truly generative model, which allows real-time generation of talking faces given speech. We have evaluated it both using multiple error measures and with a thorough test of human perception. The latter test clearly shows that our method perceptually outperforms the others and is virtually indistinguishable from reality.

References

[1] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Proceedings of ACM SIGGRAPH, Annual Conference Series*, 2003.

[2] M. Brand. Voice puppetry. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[3] C. Bregler, H. Hild, and S. Manke. Improving letter recognition by lipreading. In *Proceedings of ICASSP*, 1993.

[4] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[5] Gwenn Englebienne, Timothy F. Cootes, and Magnus Rattray. A probabilistic model for generating realistic lip movements from speech. In *NIPS*, Vancouver, December 2007.

[6] T. F. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 388–398, New York, NY, USA, 2002. ACM Press.

[7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[8] B. Theobald, G. Cawley, I. Matthews, J. Glauert, and J. Bangham. 2.5D visual speech synthesis using appearance models. *Proceedings of the British Machine Vision Conference*, 2003.

[9] Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 1999.

Self-organizing Mobile Surveillance Security Networks[1]

Duco N. Ferro Alfons H. Salden

Almende B.V.,
Westerstraat 50, 3016 DJ Rotterdam

1 Introduction

Organizations physically located in urban areas and isolated industrial zones generally hire private security companies to guard their property from theft and vandalism. In the case of mobile surveillance, a security company plans frequent by security guards to their client premises to deter and, possibly, observe inappropriate actions. This planning consist of different routes, that is, sequences of locations for each guard and the tasks, such as closing down the building, that are associated with each of these location. In addition, personnel of the security company dispatch center may act upon alarms (e.g., burglar alarms), for example, by sending a guard for further inspection. The requirements with respect to these actions (e.g., the maximum time to arrival at the location of the alarm) are specified as part of a contractual agreement.

While on patrol, a typical course of events during a work shift of a security guard does not only entail the acts of transportation and performing location specific tasks, but also frequent communication with, for example, a responder of the dispatch center or with a team leader. This communication need is particularly important if an incident occurs. Knowing whom to contact is paramount when confronted with such unexpected events in order to meet the contractual requirements. In this paper, we present an agent-based support system, ASK-ASSIST, which offers incident-specific support to guards, team-leaders and dispatch operators by recommending persons or groups (coalitions) for communication.

2 Ranking and Recommending Coalitions

The problem of grouping together individuals in an effective way has been studied in a variety of different settings. For example, two-sided matchmaking tries to bring individuals together into couples. However, a security guard, for example, needs to contact another security guard via a responder of the dispatch center and, possibly, a team leader for a particular reason (e.g., when certain expertise about a security object is required). Those extra organizational roles and context should be taken into account in the formation process, as the qualities of the coalitions heavily depend on them. Therefore, we say both actors and contextual elements belong to the coalition formation environment associated with the problem domain. In earlier work we have given a formal definition of the concept of such a coalition formation environment [2] and its application to mobile surveillance in security.

Essential to the dynamic coalition formation framework is the distinction between active elements (e.g. guards, team-leader, and dispatch operators) and passive elements (e.g., tasks, routes. security objects). These elements together appear in particular configurations. In order to recommend suitable communication coalitions, the effectiveness of reconfigurations must be evaluated and ranked. Following similar ideas presented in the literature [5] for quality of context-based optimization of end-to-end mobile-health data delivery services, this ranking can be formally captured by inducing context-dependent hierarchies of reconfigurations.

[1]The full version of this paper appeared in: Proceedings of the 2nd International Conference on Bio Inspired mOdels of NEtwork, Information and Computing Systems (BIONETICS 2007), Budapest, Hungary, December 10-13, 2007.

Collaborative content-based filtering [1] can readily be applied to incident-specific coalition formation problems in the security domain. Joint kernels weighing past reconfigurations (explicitly by measuring feedback or implicitly by measuring occurrence) can be used to predict and recommend ranked lists of reconfigurations, i.e. context-dependent hierarchies on the reconfiguration function, given not anticipated events. Also machine learning (e.g. Bayesian network classifiers) can be applied in order to obtain such hierarchies [3]. Although these approaches provide means to ground those hierarchies in actual and past contexts, they do not generate robust coalition formation solutions, when those contexts. For instance, locations of security guards and road traffic states can only be known up to a certain level of resolution. In order to arrive at robust grounded coalition formation alternatives we elaborate on how to apply a dynamic scale-space paradigm [4].

3 Simulation Experiments

An important problem in mobile surveillance is the assignment of alarms to routes. This process can be very time consuming. On a total of 12694 alarm assignments in a period of 9 months, the average assignment time is about 19 minutes. The amount of alarms that is assigned after 10 minutes is 45.6%. About 8.7 % of the alarms are assigned after one hour.

To speed up this process, we developed a functionality that allows a team-leader to set up a multi-party phone call. We varied the amount of months used for training and compared the resulting accuracy the top three suggestions made by our recommender system against the real decision made by the team-leader. We repeated these experiments for three different simple context-sensitivities. The test set concerned the alarms occurring during one month (1187 alarms).

Normally, in 59.45% of the alarm occurrences, the guard arrives in time (i.e., within 30 minutes). In our simulations, we improve the response times to alarms. Training on data from the last 6 months, 68.2% of the alarms is handled within 30 minutes. Training on data from the last 9 months, 70.78% of the alarm is handled within 30 minutes.

References

[1] Basilico, J. and Hofmann, T. (2004) Unifying collaborative and content-based filtering, Proceedings Twenty-first International Conference on Machine Learning (ICML'04).

[2] Ferro, D.N., Salden, A.H and Valk, J.M. (2007) A Robust Coalition Formation Framework for Mobile Surveillance Incident Management, In: Proceeding of the 4th Intelligent Human Computer Systems for Crisis Response and Management ISCRAM'07 (Eds. B. Van de Walle, P. Burghardt and C. Nieuwenhuis), pp. 479-488.

[3] Nurmi, P., Salden, A., Lau, S. L., Suomela, J., Sutterer, M., Millerat, J., Martin, M., Lagerspetz, E., and Poortinga, R. (2006) A system for context-dependent user modeling, in Proceedings OTM Federated Workshops, Lecture Notes in Computer Science, Vol. 4278, pp 1894–1903, Springer-Verlag.

[4] Salden, A. H., Ter Haar Romeny, B. M., and Viergever, M. A. (2001) A Dynamic Scale Space Paradigm, Journal of Mathematical Imaging and Vision 15(3), 127-168.

[5] Widya, I. A., van Beijnum, B. J. F., and Salden, A. (2006) QoC-based Optimization of End-to-End M-Health Data Delivery Services, in 14th IEEE International Workshop on Quality of Service (IWQoS'06), New Haven (CT), USA, pp 252–260, IEEE Computer Society Press.

Engineering Large-scale Distributed Auctions*

Peter Gradwell [a] Michel Oey [b] Reinier Timmer [b] Frances Brazier [b]
Julian Padget [a]

[a] *University of Bath*
[b] *Vrije Universiteit Amsterdam*

Abstract

The functional characteristics of market-based solutions are typically best observed through the medium of simulation, data-gathering and subsequent visualization. We previously developed a simulation of multiple distributed auctions to handle resource allocation (in fact, bundles of unspecified goods) and in this paper we want to deploy an equivalent system as a distributed application. There are two notable problems with the simulation-first, application-second approach. First, the simulation cannot reasonably take account of network effects. Second, how can one recreate in a distributed application the characteristics demonstrated by the mechanism in the simulation. We describe: (i) the refactorings employed in the process of transforming a uni-processor lock-step simulation into a multi-processor asynchronous system, (ii) some preliminary performance indicators, and (iii) some reflections on our experience which may be useful in building MAS in general.

The combinatorial auction (CA) is capable of delivering the optimal solution to a resource allocation problem. Unfortunately, solving combinatorial auctions is costly, and there are circumstances that make combinatorial auctions inappropriate: (i) if resources and bidders are distributed, the centralization intrinsic to a combinatorial auction may be problematic, (ii) under soft real-time constraints, an anytime (sub-optimal) algorithm may be preferable to an optimal algorithm with an unpredictable runtime, (iii) the single-point-of-failure intrinsic to combinatorial auctions may pose an unacceptable risk for system resilience.

In an earlier paper [2], we reported on the economic characteristics and run-time performance of a market-based approach in comparison with a CA, when both are applied to common, standard data sets. In this market-based approach, resources are traded in *continuous double auctions*. There is an auction for each type of resource traded. Together they form a *Multiple Distributed Auction* (MDA). Market *traders* subscribe to one or more auctions in order to sell or buy a bundle of resources, depending on whether they are *sellers* or *buyers* of resources.

In this paper, we present the issues that have arisen in refactoring a uni-processor (centralized) simulation of an MDA into a distributed agent application (using the AgentScape platform). There are two challenges in achieving this transition:

- **Concurrency:** how to introduce *just enough* concurrency to give an advantage but not so much as might paralyze or lead to significant numbers of delicate timing bugs. We refactor the original synchronous simulation a step at a time in order to constrain the available concurrency.
- **Communication:** distributed systems usually embody significant communication overheads. Careful placement of processes and resources helps to exploit locality. In addition, we are able to determine the impact of additional messaging overhead required to facilitate a distributed architecture.

The centralized MDA simulation uses a lock-step model, in which at every step (or round) the following three sub-steps are performed sequentially:

(i) All traders are instructed to check the status of their current bundle. If the bundle is still in progress, nothing is done. If the bundle has been completed or has failed (the trader has given up), they acquire a new sell or buy bundle.

(ii) All auctions are instructed to perform one round. A round consists of asking all participating traders to send a shout (a bid or an ask). Any matches will be reported to the corresponding traders.

*The full version [1] of this paper appeared in the Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08).

(iii) All traders are instructed to get any trade results. At this point, the statistics are updated with completed or failed bundles.

In the centralized simulation, a single thread of execution runs each of these rounds sequentially. However, in a distributed system, processes that have no direct effect on each other can often be executed in parallel. In the case of the MDA simulation, each auction could in principle run in parallel, which could lead to performance improvements. However, in order for the results of the distributed MDA to be comparable to the centralized one, the notion of rounds as described above must still be kept. Fortunately, within each round, we can utilize parallelism to improve performance as, *within* a step, actions can be executed in parallel. In other words, in step one, all traders can check their progress simultaneously; in step two, all auctions can run one round in parallel; and in step three, all traders can process the trade results in parallel.

The distributed implementation uses AgentScape [3], a framework for heterogeneous, mobile agents, as a base. In an agent system, typically the work is divided among several agents, which all perform a part of the work. AgentScape can distribute agents over multiple hosts, and thus spread the load of an application. Porting the centralized MDA simulation to run on AgentScape was relatively straightforward. Each auction and each trader was turned into an agent. AgentScape could then run these agents on different hosts, allowing for parallel execution.

Unfortunately, the distributed simulation ran a lot slower than the centralized simulation. This performance loss was mainly due to the communication overhead between auctions and traders, and the need to synchronize the agents per round. Further analysis of the system resulted in three optimizations, which all aimed at reducing the amount of communication (e.g., method invocations) between agents:

(i) Cache communication results, whenever possible.
(ii) Group multiple method invocations that are often called in sequence into a single invocation.
(iii) Localize communication by placing agents that communicate with each other a lot on the same host.

As with any distributed application, synchronization is not problematic if the cost of doing so is relatively small compared to the amount of work that can be done in parallel. In the MDA application, however, the average time for an auction to process a shout is much less than the time it takes the trader to send the shout to the auction and receive back the results. A larger grain size would reduce the impact of the communication overhead. This could be obtained by allowing traders and auctions to perform multiple steps at once.

In conclusion, MAS software development is characteristically evolutionary and a common starting point is a proof-of-concept system running on a single machine utilizing an agent platform or even a simulation framework. The software engineering challenge lies in how to scale that demonstrator up into a system comprising many more agents running over multiple machines. We have transformed a centralized simulation of a Multiple Distributed Auction into a multi-agent system, supporting large numbers of agents participating in large numbers of auctions on distributed machines. The AgentScape mobile agent platform was used to distribute the entities in the MDA over multiple hosts and to provide the necessary communication between these entities. Unfortunately, experiments have shown that the communication overhead of the distributed market is quite large compared to the benefit of gaining more computing resources.

The lesson learnt here is that synchronizing agents is very time consuming due to the amount of messaging involved. However, the barrier synchronization can be removed, at the cost of a more complex refactoring of the original code, and at the cost of producing *similar* but not identical results to the original code. Thus, the process reported here has been a tedious but necessary step to demonstrate functional *equivalence* before we move on to a situation in which we must define and demonstrate functional *similarity*.

References

[1] P. Gradwell, M. A. Oey, R. J. Timmer, F. M. T. Brazier, and J. Padget. Engineering large-scale distributed auctions. In *Proceedings of the Seventh Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, May 2008.

[2] P. Gradwell and J. Padget. A comparison of distributed and centralised agent based bundling systems. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*, pages 25–34, New York, NY, USA, 2007. ACM Press.

[3] B. J. Overeinder and F. M. T. Brazier. Scalable middleware environment for agent-based Internet applications. In *Applied Parallel Computing*, volume 3732 of *Lecture Notes in Computer Science*, pages 675–679. Springer, Berlin, 2006.

A Cognitive Model for the Generation and Explanation of Behavior in Virtual Training[1]

Maaike Harbers [ab] Karel van den Bosch [b] Frank Dignum [a]
John-Jules Meyer [a]

[a] *Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands*
[b] *TNO Defence, Security & Safety, Kampweg 5, 3796 DE Soesterberg, The Netherlands*

1 Introduction

Virtual systems have become common training instruments in organizations such as the army, navy and fire services. In most of the current systems instructors play a major role; they enter the effects of trainee actions into the system, play the role of other characters in the scenario, and provide instruction, guidance and feedback to the trainee. Cognitive models can be used to support instructors by autonomously generating virtual character behavior. However, without knowing the reasons for an action, it is more difficult for a trainee to understand the situation and learn from it. Some virtual training systems with explanation components have been proposed, e.g. Debrief [3] and the XAI system [5, 1]. These components explain a character's actions by giving information about its state at the time it performed an action. However, the explanations do not provide the character's underlying motivations for its actions.

We propose a cognitive model based on BDI theory (belief desire intention) that is able to generate character behavior and explanations. Because the character reasons and plans with BDI concepts, the model is also able to generate explanations in terms of the characters's beliefs and goals. Such explanations in high level concepts are easy to understand for humans. Moreover, they not only provide information about the character's state, but yield insight in the reasons why it performed a certain action.

2 The Cognitive Model

Our approach resembles planning methods based on hierarchical task networks (HTNs) [4]. In HTN planning, an initial plan describing the problem is a high-level description of what is to be done. Plans are refined by action decompositions, which reduces a high-level action to a set of lower-level actions. Actions are decomposed until only primitive actions remain in the plan. In our model, initial plans, action decompositions and primitive actions correspond to main goals, divisions of goals into sub-goals and actions, respectively. Figure 1 represents the cognitive model of a simple virtual fire-fighter.

The virtual characters in a training scenario fulfil specific roles with corresponding goals and tasks. For example, a dispatch center operator should properly inform others, a policeman ensure order and safety, and a fire-fighter bring incidents to a successful conclusion. The characters maintain their overall goal during the complete training session. To achieve their main goal, they have to adopt proper sub-goals, sub-sub-goals, and finally perform the corresponding actions. The selection of goals depends on a character's beliefs and the relation between a goal and its sub-goals. Relations differ for example on the amount of sub-goals to be achieved in order to achieve a goal (one or all), or the order in which sub-goals have to be achieved (fixed or free order). The fire-fighter in our example has the beliefs **Fire** and **Victim**. Because it also has the belief

[1]The full version of this paper will appear in the proceedings of ExaCt 2008, ECAI Workshop on Explanation Aware Computing. This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

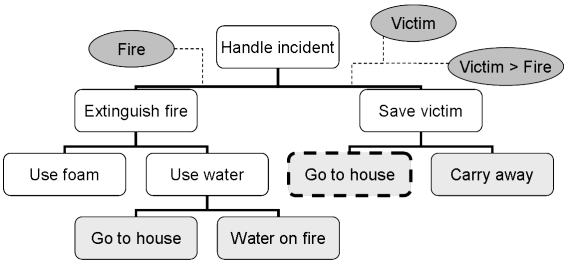


Figure 1: The cognitive model of a fire-fighter

that extinguishing a fire has priority over saving a victim (**Victim** $>$ **Fire**), it adopts the goal **Save victim** and executes the action **Go to house**.

For the generation of behavior, the cognitive model is used from top to bottom and the result is a sequence of actions. For the generation of explanations, the model is used from actions to the underlying sub- and main goals. In an explanation, the beliefs and goals that were responsible for a particular action are made explicit. However, a long trace of beliefs and goals may underly one action, but a trainee often does not need all information. A selection mechanism is needed to filter the most useful information out of the many goals and beliefs. A possible heuristic is that an action which is part of different goals in the task hierarchy is explained by the beliefs that were responsible for the selection of the current goal and not the other. For example, the action **Go to house** in Figure 1 can be part of two goals and is explained by the character's beliefs **Victim** and **Victim** $>$ **Fire**.

The implementation of the cognitive model requires an agent programming language in which beliefs and goals are explicitly represented. Moreover, because explanations often involve goals *and* beliefs, it should be possible to combine the two and reason with them. In the current BDI-based agent programming languages, agents cannot reason about their own goals and beliefs. This has been avoided because modifications to beliefs and goals in an agent's reasoning process might result into undesired loops. We have chosen to implement our model in 2APL [2] and make updates of the agent's goals and beliefs in its belief base. The updated goals and beliefs are strictly separated from the 'normal' beliefs and do not influence the generation of actions. They are only used for the generation of explanations.

3 Conclusion

We have proposed a cognitive model for the generation and explanation of behavior. The use of explicit goals and beliefs in an agent's reasoning process distinguishes our model from most other approaches of behavior generation. Moreover, explanations generated by other accounts of explaining agents do not refer to the agent's beliefs and goals.

References

[1] M.G. Core, T. Traum, H.C. Lane, W. Swartout, J. Gratch, and M. van Lent. Teaching negotiation skills through practice and reflection with virtual humans. *Simulation*, 82(11), 2006.

[2] M. Dastani. 2APL: a practical agent programming language. *Autonomous Agents and Multi-agent Systems*, 16(3):214–248, 2008.

[3] W. Lewis Johnson. Agents that learn to explain themselves. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 1257–1263, 1994.

[4] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Pearson Education, Inc., New Jersey, USA, second edition, 2003.

[5] M. Van Lent, W. Fisher, and M. Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. In *Proceedings of IAAA 2004*, Menlo Park, CA, 2004. AAAI Press.

Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning[1]

Koen Hindriks Dmytro Tykhonov

Man-Machine Interaction Group,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft

1 Introduction

In bilateral negotiation, two parties aim at reaching a joint agreement. They do so by exchanging various offers or bids using e.g. an alternating offers protocol [2]. In reaching such an agreement both parties usually aim to satisfy their own interests as best as possible, but have to take their opponent's preferences into account as well to reach an agreement at all. This is complicated by the fact that negotiating parties are generally not willing to reveal their preferences in order to avoid exploitation. As a result, both parties have incomplete information which makes it hard to decide on a good negotiation move and hard to reach an optimal agreement.

In this paper, we show that it is nonetheless possible to construct an *opponent model*, i.e. a model of the opponent's preferences that can be effectively used to improve negotiation outcomes. We provide a generic framework for learning both the preferences associated with issue values as well as the weights that rank the importance of issues to an agent. The main idea is to exploit certain structural features and rationality principles to guide the learning process and focuses the algorithm on the most likely preference profiles of an opponent. We present a learning algorithm based on Bayesian learning techniques that computes the probability that an opponent has a particular preference profile. Our approach can be integrated into various negotiating agents using different strategies.

2 Learning an Opponent Model

Our goal is to introduce a learning approach that can be used to model an opponent in a negotiation with imperfect information. First, we define a hypothesis space that specifies the range of opponent profiles that are considered. We do so by introducing various reasonable assumptions about the structure of opponent profiles as well as about an opponent's negotiation strategy.

Our first assumption is a common one, see e.g., [2], and assumes that the utility of a bid can be computed as a weighted sum of the utilities associated with the values for each issue. Utility functions modelling the preferences of an agent thus are linearly additive functions and are defined by a set of weights (or priorities) and corresponding evaluation functions for each of n issues. In order to learn an opponent's preference profile or utility function we need to learn both the issue weights as well as the evaluation functions. The objective of learning an opponent model thus is to find a model that is the most plausible candidate or best approximation of the opponent's preference profile.

Our next assumption concerns the issue priorities in a preference profile. We define the set of hypotheses about the private weights of an opponent as the set of all possible rankings of weights. It is then straightforward to associate real-valued numbers again with a hypothesis about weights, which can be computed as a linear function of the rank. Finally, we need to impose some additional structure on the evaluation functions

[1] The full version of this paper appeared in: *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16, 2008, Estoril, Portugal.*

in order to be able to learn a preference profile. To facilitate the learning of an opponent's preferences over issue values we introduce a hypothesis space of predefined function shapes (*downhill, uphill, and triangular*). To summarize, the set of hypotheses concerning an opponent's preference profile is a Cartesian product of the hypotheses about issue weights and shapes of issue evaluation functions.

The idea is to learn an opponent preference profile from its negotiation moves, i.e. the bids it proposes during a negotiation. In a Bayesian learning approach, this means we need to be able to update the probability associated with all hypotheses given new evidence, i.e. one of the bids. More precisely, we need to compute the probability of every hypothesis given the bid proposed by the opponent. In order to be able to use Bayes' rule to do this, however, we need some information about the utility that the opponent associates with a bid. To this end it is assumed that an agent's tactics during a negotiation can be defined by a monotonically decreasing function [4]. This assumption is reasonable given that negotiating agents have to concede to reach an agreement and still allows the use of various kinds of tactics since no exact knowledge about an opponent's negotiation tactics is assumed. More specifically, the rationality assumption is modelled as a probability distribution associated with a range of tactics. As a result, each utility associated with an opponent's bid thus also has an associated probability. Finally, during a negotiation an agent can use the updated probability distribution to compute expected utility of counteroffers it considers and choose one that e.g. maximizes the utility of its opponent, to increase the likelihood of acceptance by that opponent.

Experiments have been performed to show the effectiveness of our approach to learn the opponent model and to use it to find a good counteroffer (see the full paper). The Bayesian learning agents used in the experiment update their opponent model each time a new bid is received from the opponent in line with the Bayesian learning approach introduced above. The strategy used by the Bayesian learning agents is based on the smart meta-strategy of [1]. The agent starts with proposing a bid that has maximal utility given its own preferences. Each consecutive turn the agent can either accept the opponent's bid or send a counter-offer. The agent accepts a bid from its opponent when the utility of that bid is higher than the utility of its own last bid or the utility of the bid it would otherwise propose next. Otherwise, the agent will propose a counter-offer. In this domain, the Bayesian agents very efficiently learn issue weights when they are provided with domain knowledge, indicated by the fact that the negotiation trace almost coincides with the Pareto frontier. But even without domain knowledge the Bayesian agent needs little time to learn the issue evaluation functions and consecutively improves the weight estimations. The influence of the negotiation domain, preference profile, and opponent's strategy on the quality of learning was investigated in [3].

3 Conclusions

In this paper, an opponent modelling framework for bilateral multi-issue negotiation has been presented. The main idea proposed here to make opponent modelling in negotiation feasible is to assume that certain structural requirements on preference profiles and on the strategy of an opponent are in place. Due to the probabilistic nature of the model, these assumptions still allow for a great diversity of potential opponent models. The learning approach has been tested on several domains to demonstrate the effectiveness of the approach. The results moreover showed the effectiveness of using an opponent model in a negotiation strategy to improve the efficiency of the bidding process. In future work we will analyze the quality of the learned opponent model with respect to the original preferences profile of the opponent.

The learning approach does not rely on prior knowledge about e.g. the domain, but if such knowledge is available it can be incorporated and used to initialize probability distributions in the opponent model. Domain knowledge would also be useful to increase the efficiency of learning a correct opponent model in learning algorithm proposed (for more details see the full paper).

References

[1] Faratin, P., Sierra, C., and Jennings, N. R. 2003. Using Similarity Criteria to Make Negotiation Trade-Offs, Journal of Artificial Intelligence, 142 (2), pp. 205-237.

[2] Raiffa, H. *The Art and Science of Negotiation*, Harvard University Press. 1982.

[3] Hindriks, K.V., Tykhonov, D. 2008. Towards a Quality Assessment Method for Learning Preference Profiles in Negotiation, in Proc. of AMEC'08.

[4] Zeng, D., and Sycara, K. 1998. Bayesian Learning in Negotiation, International Journal of Human Computer Systems, vol. 48, pp. 125-141.

Exploring Heuristic Action Selection in Agent Programming [1]

Koen V. Hindriks[a] and Catholijn M. Jonker[a] and Wouter Pasman[a]

[a] *Delft University of Technology, P.O.Box 5031, 2600 GA, Delft*

Abstract

Rational agents programmed in agent programming languages derive their choice of action from their beliefs and goals. One of the main benefits of such programming languages is that they facilitate a high-level and conceptually elegant specification of agent behaviour. Qualitative concepts alone, however, are not sufficient to specify that this behaviour is also nearly optimal, a quality typically also associated with rational agents. Optimality in this context refers to the costs and rewards associated with action execution. In this paper we extend the agent programming language GOAL with primitives that allow the specification of near-optimal behaviour and illustrate the use of these constructs by extending a GOAL Blocks World agent with various strategies to optimize its behaviour.

1 Introduction

We use the well-known Blocks World domain [**?**] to explore and present evidence for the usefulness of adding expressive programming constructs that allow the specification of utility-based heuristic strategies for action selection to the agent programming language GOAL [**?**]. By means of various examples we illustrate that the new constructs introduced allow for an elegant specification of such strategies. Additionally, we present some experimental results that demonstrate the usefulness of the programming constructs.

Our objectives are twofold: (i) to extend GOAL with programming constructs to define a heuristic or utility-based decision capability as an additional action selection mechanism. Such constructs allow the optimization of agent behaviour as well as reduce the amount of nondeterminism present in an agent program, and (ii) to assess the usefulness of the mechanism by comparing the behaviour of a GOAL agent which does not use the mechanism with various instantiations of GOAL agents that do use it.

Research in planning has shown that in order to plan effectively and be able to generate near-optimal plans for the Blocks World it must be possible to specify various domain-dependent heuristics [**?**]. The specification of these heuristics in domain-independent planning systems requires the right concepts to express and implement them. If agent programming languages are to match these capabilities, programming constructs with similar expressive power need to be available to program rational agents that use heuristics to improve performance. We argue that in programming languages for rational agents such programming constructs would be most useful if they allow for the specification of such heuristics in terms of the core concepts of beliefs and goals present in these languages.

Agent programming languages in general, and GOAL in particular, quite naturally lead to writing programs that are underspecified (i.e. such programs do not determine a unique action to perform next and thus may underspecify the actual behaviour of an agent). The operational semantics of these languages leaves room for various alternatives as to how to implement the action selection mechanism of an agent. One of the benefits of underspecification is that it facilitates and simplifies the design and programming of an agent, but it may also give rise to suboptimal behaviour (due to ad hoc suboptimal choices). The idea is to introduce another, utility-based mechanism for action selection on top of the qualitative selection mechanism already present in GOAL that can be used to further limit the number of choices.

We introduce a generic extension of the GOAL agent programming language that can be incorporated into other agent languages based on concepts of belief and goal, and add a capability for specifying heuristic

[1] Full paper accepted for publication in the Proceedings of the 6th Workshop on Programming Multi-Agent Systems, Springer

selection strategies by means of utility functions. We introduce a programming construct for specifying utility values and define a semantics for the extension of GOAL with a utility-based action selection mechanism. The programming constructs we introduce allow for an elegant specification of behaviour that shows improved performance compared with a GOAL agent that does not make use of the utility-based selection mechanism.

The basic idea is to associate a quantitative number with the execution of an action a in a state m, i.e., to associate a real valued number $U(m, a, m')$ with executing a in state m resulting in a new state m'. The main difference with other approaches is that the states m are mental states of an agent that consist of its beliefs and goals, declaratively specified in some basic language such as that of propositional logic. A number associated with an action in this way can be perceived of in two different ways. One perspective, the more principled view on what this number represents, is to suggest that the number is a utility value that represents how much value is to be gained from executing the action. It is standard to further decompose such a utility value into two components, a *cost component* that is associated with taking an action in the starting state and a *reward component* that associates a reward with getting to the resulting state (cf. [**?**]). Alternatively, such a number can be perceived of as a heuristic that only provides an estimation of e.g. the costs of executing an action.

In order to incorporate the assignment of quantitative values and associate these with transitions of a GOAL program, such programs are extended with a new utility section and the following notation is introduced for representing utility:

```
value(<initial-state-cond>, <action-descr>, <successor-state-cond>) = <utility-expr>
cost(<initial-state-cond>, <action>) ≝ -1·value(<initial-state-cond>, <action>, true)
reward(<successor-state-cond>) ≝ value(true, any, <successor-state-cond>)
```

The `initial-state-cond` as well as the `successor-state-cond` refer to arbitrary mental state conditions, i.e., conditions that are combinations of `a-goal(...)` and `bel(...)` operators. In addition, the constant `true` - which holds in any mental state - may be used here as well. The `action-descr` part refers to any action description that is allowed in GOAL, e.g., in the Blocks World `move(X,Y)`. Variables are allowed in both mental state conditions used to characterize the initial or resulting state, as well as in the action description. The same holds for the `utility-expr` part, which denotes a numerical expression which may involve basic arithmetic operators such as addition and multiplication. The action description part may also be filled with a special don't care label `any`.

As an example, we consider heuristics to improve performance in the Blocks World domain. A simple idea to improve performance here is to give priority to so-called *constructive* moves over other moves [**?**]. Such moves put a block in place thus bringing the current state closer to the goal state, and in addition may make it possible to perform another constructive move next. Using the `cost` construct to assign costs to actions we can make sure that a constructive move always has an associated cost less than that for other types of moves. By means of the mental state condition `bel(tower([X|T])), a-goal(tower([X,Y|T]))` we can pick out moves that are constructive, and we can define the cost function we are looking for as follows:

```
case{
  bel(tower([Y|T]), a-goal(tower([X,Y|T])): cost(move(X,Y)) = 1.  % a constructive move
  true:                                     cost(move(X,Y)) = 2.  % otherwise
}
```

We have introduced new programming constructs that add expressiveness to the GOAL programming language and allows to specify utility-based heuristics using high-level concepts such as beliefs and goals. The construct can be added to any programming language that is based on these agent concepts. Thus, high-level agent programming concepts are combined naturally with a utility-based action selection capability.

References

[1] Fahiem Bacchus and Froduald Kabanza. Using Temporal Logics to Express Search Control Knowledge for Planning. *Artificial Intelligence*, 116(1-2):123–191, 2000.

[2] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94, 1999.

[3] F.S. de Boer, K.V. Hindriks, W. van der Hoek, and J.-J.Ch. Meyer. A Verification Framework for Agent Programming with Declarative Goals. *Journal of Applied Logic*, 5(2):277–302, 2007.

[4] John Slaney and Sylvie Thiébaux. Blocks World revisited. *Artificial Intelligence*, 125:119–153, 2001.

Individualism and Collectivism in Trade Agents (Extended Abstract)

Gert Jan Hofstede[a] Catholijn M. Jonker[b] Tim Verwaart[c]

[a] *Wageningen University, Postbus 9109, 6700 HB Wageningen*
[b] *Delft University of Technology, Mekelweg 4, 2628 CD Delft*
[c] *LEI Wageningen UR, Postbus 29703, 2502 LS den Haag*

Abstract

This paper was originally presented at IEA/AIE 2008 [1]. The current paper is an extended abstract for presentation at BNAIC 2008. Agent-Based Modeling can contribute to the understanding of international trade processes. Models for the effects of culture and cultural differences on agent behavior are required for realistic agent-based simulation of international trade. This paper makes a step toward modeling of culture in agents. The focus is one of the five dimensions of culture according to Hofstede: individualism versus collectivism. The paper presents an analysis based on social science literature about national cultures. For cultural differentiation of agent behavior, rules are formulated for individualist versus collectivist agent behavior with respect to negotiations, cooperation or defection in the delivery phase of transactions, trade partner selection, and trust. Example computations demonstrate the feasibility in multi-agent simulations.

1 Representation of Individualism and Collectivism in Trade Agents

Agent-Based Economics (ABE) studies economic processes as interactions of individual actors. Cultural differences are known to have their effects on international business interactions and on trust between business partners. Models of culture-bound agents will advance the understanding through ABE of intercultural trade processes as well as differences in trade processes across cultures. Culture has different aspects or dimensions [2]. The current paper focuses on the widely recognized distinction between individualistic and collectivistic cultures. It presents an exercise in multi-agent simulation, with the purpose of better understanding the mechanisms that promote or hinder international trade.

People are gregarious by nature. But the life conditions of societies vary, and they have adapted accordingly. Hunter-gatherers live in small bands, usually consisting of a few nuclear families. In agricultural societies, larger units have developed, and the people may live in extended families or clans. This is still the default model of social organization in most of the world, although it is being put under strain by urbanization. In modern, affluent industrial societies people tend to revert to nuclear families. The variation in basic group size and cohesion between societies has been shown by sociologists, e.g., in the distinction between *Gemeinschaft* and *Gesellschaft* that Tönnies introduced as early as 1887. In a *Gemeinschaft*, people share everything, both material and immaterial, whereas in a *Gesellschaft*, private property and other individual-centered institutions are possible. This variation has been confirmed by social psychological cross-national studies of practices or values, for instance the work of Triandis [3] and Hofstede [2]. These authors speak of the distinction between individualism and collectivism. Table 1 shows some typical distinctions. With respect to geographical distribution, national cultures of East-Asia and Central America are extremely collectivistic, while the Anglo countries are at individualistic extreme of the scale.

On the basis of Hofstede's theory, the paper formulates the expected influence of individualism versus collectivism on the behavior of traders in the processes of trade goal selection, trade partner selection, bargaining, and delivery and trust, as well as the maintenance of beliefs about potential trading partners. Subsequently, the verbal models of culturally differentiated behavior are formalized into production rules that take culture and personal traits of the agents into account.

Table 1. Some distinctions between norms in collectivist and individualist societies (source: Hofstede [2]).

Collectivist	Individualist
Maintain harmony, avoid confrontation	Speak your mind
High-context, implicit communication	Low-context, explicit communication
Use the word "we"	Use the word "I"
Show favor to in-group customers	Treat all customers equally
No business without a personal relation	Task is more important than a good relation
A relation brings rights and obligations	Mutual advantage is the basis of relations
Relations are given	Build and maintain relations actively
Save face for in-group	Keep self-respect
Responsible for group interests	Responsible for personal interests

2 Simulation Results and Conclusion

Table 2 presents some results of multi-agent simulations. Simulations in populations with collectivistic agents belonging to different groups typically show the distribution of run 1 in Table 2: in-group trade. Individualist agents rapidly develop networks of preferred relations, on which they trade very efficiently. In mixed settings like run 2, the individualists develop the same pattern, but collectivist agents stick to their in-group trade. However, in run 3 where no in-group partners are available, the collectivist agents develop the individual relations pattern. In run 4 only group C agents can find in-group partners and show the collectivist pattern, while the other collectivist agents develop the individual relations pattern.

The work presented in this paper shows that the approach to simulate cultural dependent behavior in agents, leads to behavior that corresponds to human behavior in trade simulation games. Therefore, the paper shows that agent-based simulation contributes to the understanding of international trade processes.

Table 2. Number of successful transactions in simultations with 8 suppliers and 8 customers.

1. Customers		collectivist gr *A*				collectivist gr *B*			
Suppliers		C1	C2	C3	C4	C5	C6	C7	C8
collectivist	S1	9	7	5	3	0	0	0	0
group *A*	S2	3	8	9	3	1	0	0	0
	S3	5	6	5	4	0	0	0	0
	S4	2	3	5	11	0	0	0	0
collectivist	S5	0	0	0	0	4	5	5	4
group *B*	S6	0	0	0	0	6	6	5	5
	S7	0	0	0	0	8	4	6	6
	S8	0	0	0	0	8	7	8	9

2. Customers		individualist				collectivist gr *A*			
Suppliers		C1	C2	C3	C4	C5	C6	C7	C8
individualist	S1	3	2	24	0	0	0	0	0
	S2	1	0	0	12	0	1	0	1
	S3	0	16	1	2	0	0	0	0
	S4	14	2	0	7	0	0	0	0
collectivist	S5	0	0	0	0	5	6	6	6
group *A*	S6	0	0	0	0	8	3	5	6
	S7	0	0	0	0	5	7	5	8
	S8	0	1	0	0	8	7	5	4

3. Customers		individualist				collectivist gr *A*			
Suppliers		C1	C2	C3	C4	C5	C6	C7	C8
individualist	S1	1	25	1	0	0	0	0	0
	S2	9	1	0	5	1	0	3	6
	S3	0	0	2	0	0	0	9	4
	S4	2	0	0	1	15	3	5	0
collectivist	S5	1	0	0	0	0	20	0	0
group *B*	S6	0	1	1	17	0	0	4	0
	S7	5	0	4	0	1	0	1	9
	S8	1	0	18	0	1	0	1	1

4. Customers		collectivist gr *C*				collectivist gr *B*			
Suppliers		C1	C2	C3	C4	C5	C6	C7	C8
collectivist	S1	6	7	7	5	0	0	0	0
group *C*	S2	9	3	4	8	0	0	0	0
	S3	4	7	6	6	0	0	1	0
	S4	4	5	5	6	0	0	0	0
collectivist	S5	0	0	0	0	17	0	1	1
group *A*	S6	0	0	0	0	1	0	1	16
	S7	0	0	0	0	0	23	1	1
	S8	0	0	1	2	1	15	1	2

References

[1] G.J. Hofstede, C. Jonker, T. Verwaart Individualism and collectivism in trade agents. In Nguyen, N.T. et al., editors, *New Frontiers in Applied Artificial Intelligence, Proceedings of IEA/AIE 2008*, Springer-Verlag, Berlin Heidelberg, Lecture Notes in Artificial Intelligence 5027: 492-501, 2008.

[2] G. Hofstede. *Culture's Consequences, second edition*. Sage Publications, Thousand Oaks CA, 2001.

[3] H.C. Triandis. *Individualism and collectivism*. Westview, Boulder CO, 1995.

Agents Preferences in Decentralized Task Allocation (extended abstract) [1]

Mark Hoogendoorn [a, b] Maria L. Gini [b]

[a] *Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, email: mhoogen@cs.vu.nl*

[b] *University of Minnesota, Department of Computer Science and Engineering, 200 Union Street SE, Minneapolis, MN 55455, United States of America, email: gini@cs.umn.edu*

1 Introduction

Auctions are used in multi-agent systems, among other things, to perform allocation of tasks (see e .g. [2] and [3]). Such reverse auctions, where the buyer is the auctioneer, can be of a combinatorial type, allowing for bidding on bundles of tasks. Sandholm [1] notes that reverse auctions are not economically efficient because optimal bundling depends on suppliers preferences, which traditionally cannot be expressed. Enabling the agents to express the preferences of their users is an important requirement for actual companies and people to use agents for bidding.

In this paper we propose a concrete preference function to be used by an agent to express preferences over tasks. This function expresses preferences for specific properties of tasks namely (1) duration, (2) task type, and (3) start and end time. The function is used in a decentralized task allocation setting. We introduce a bidding algorithm, where an agent bids on its most preferred tasks that are feasible given its current commitments. This algorithm uses a pricing mechanism which depends on the actual cost to perform the tasks and on the preference for the task. The influence of preferences on the price can be varied by setting a parameter.

Using this algorithm, we investigate the impact of preferences upon other aspects of task execution, such as execution time. We use both synthetic as well as real data from a logistics company.

Below, a brief overview of the approach is given, including the results using the synthetic and the real life dataset.

2 Bidding using Preferences

Preferences in our case can be a combination of the following: (1) a preference for tasks of a particular duration (e.g. I hate performing very short tasks), (2) a preference for tasks at particular times during the day (e.g. I love getting up early in the morning, so give me tasks that ought to start early in the morning), and (3) a preference for particular types of tasks (e.g. I really hate to perform a task like that). Hereby, each of these preferences can be specified in a natural way, which is mapped (using particular functions) to a real number on the interval [0,1]. Given these three ingredients of the preference function, a weighed sum is taken, resulting in a preference value for the specific task (ϕ_{task}). In the bidding algorithm we propose, price is used as a mechanism to express these preferences for tasks.

The bidding algorithm proposed starts when a request for quotes (RFQ) arrives. The tasks within this RFQ are then ordered based upon their preference. If some tasks have identical preferences, they are ordered

[1] The full version of the paper appeared in: M. Ghallab, C.D. Spyropoulos, N. Fakotakis and N. Avouris (eds.), Proceedings of the 18th European Conference on Artifi cial Intelligence, ECAI'08. IOS Press, 2008, pp. 398-402.

according to the start time specified in the RFQ for the tasks included. We assume that there exists a function *switch_time:* TASK_TYPE $\times$ TASK_TYPE $\rightarrow$ DURATION that calculates the switching time from one task type to another (when it can be performed on the resource). Furthermore, *performance_time:* TASK_TYPE $\rightarrow$ DURATION expresses the time needed to perform the task.

Bidding Algorithm

For each preference ordered task:
Check if task (*current*) can be done using the resource.
If yes, see if it fits in the current schedule (see below).
From the beginning of the schedule and for each empty slot
in the schedule do:
If the task fits in the current empty slot in the schedule
then insert the task in the bid,
add its time parameters to the schedule, and
compute the price of the bid (see below)
else if *latest_end_time*$_{current}$ > *latest_end_time*$_{next}$
then continue with the next slot
else continue with the next task.
The price of the bid is computed as follows (note the parameter *p*):
price$_{task}$ =
$(1 + (p \times (1 - \phi_{task}))) \times$
[switch_time(type$_{previous}$*, type*$_{current}$*)* +
$switch\_time(type_{current}, type_{next},)$ +
$performance\_time(type_{current})]$

3 Results

We evaluated our approach in two ways, first by rigorously testing it with synthetic data. Several parameters have been varied, namely the tightness of the time windows in which the tasks need to be performed and the relative availability of resources. It was shown that it was easiest to get preferences awarded for markets with wide time windows. The trade-off between meeting preferences and overall execution time has been studied in depth. We have shown that the overall execution time is influenced most in the case of the overflow market, due to the fact that in the shortage market there are hardly any alternatives at hand and therefore, although the agent might not prefer a task, it will still get its bid awarded. The curves observed tend to have the same shape when the time window setting changes but the market type remains the same. For different market types, the curves vary in steepness.

Besides testing with synthetic data, we have also used a real company dataset from the trucking domain. We have shown that the bidding algorithm is effective in awarding suppliers more preferred tasks. The influence of this preference on the overall solution quality was not observed using the real dataset. Hence, in this setting the preferences being met have much less influence on the efficiency of the solution found.

Acknowledgments: Partial support is gratefully acknowledged from NSF under grant IIS-0414466.

References

[1] Tuomas Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–58, Fall 2007.

[2] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Trans. Computers*, 29(12):1104–1113, December 1980.

[3] William Walsh and Michael Wellman. A market protocol for decentralized task allocation and scheduling with hierarchical dependencies. In *Proc. of 3th Int'l Conf on Multi-Agent Systems*, 1998.

Agent-based Patient Admission Scheduling in Hospitals

Anke K. Hutzschenreuter[a b] Peter A. N. Bosman[b] Ilona Blonk-Altena[c]

Jan van Aarle[c] Han La Poutré[a b]

[a] *Faculty of Technology Management, Eindhoven University of Technology, The Netherlands*, a.k.hutzschenreuter@tue.nl

[b] *Center for Mathematics and Computer Science, Amsterdam, The Netherlands*

[c] *Dept. of Planning and Management, Catharina Hospital Eindhoven, The Netherlands*

The full version of this paper has been published in the proceedings of the *2008 AAMAS* conference.

1 Introduction

Scheduling decisions in hospitals are often taken in a decentralized fashion. This means that different specialized hospital units decide autonomously on patient admissions or operating room schedules. We present an agent-based model for the selection of an optimal mix for patient admissions. Admitting the right mix of patients is important in order to optimize the resource usage and patient throughput. Our model is based on an extensive case analysis, involving data analysis and interviews, conducted in a case study at the Catharina Hospital Eindhoven (CHE). We focus on the coordination of different surgical patient types[1] with probabilistic treatment processes involving multiple hospital units. We also consider the unplanned arrival of other patients requiring (partly) the same hospital resources. Simulation experiments show the applicability of our agent-based decision support tool. The simulation tool allows for the assessment of resource network usage as a function of different policies for decision making. Furthermore, the tool incorporates a first optimization module for the resource allocation of postoperative care beds.

2 Approach

The agent-based simulation system for admission scheduling comprises two types of agent: OR scheduling agents and resource agents. OR scheduling agents are responsible for managing the operating room schedule of a surgical specialty. This comprises the scheduling of admitted patients to operating room (OR) time slots according to the scheme. Also, the agent requests patient admissions for the following day's OR scheme and sends requests for postoperative transfers. Resource agents act on behalf of a postoperative or critical care hospital unit and coordinate patient admissions, transfers and discharges with other agents. In general, patients are admitted to a hospital unit only if beds are available. If no bed is available, some agents may use back-up capacity which is accounted for the in the system's performance. If no regular bed is free and no back-up capacity is available at the indicated hospital unit, a resource agent may approach a resource agent of higher care level for transfer. A higher care level indicates a higher intensity of care and monitoring.

The sequence of required treatment operations to be performed at the different hospital units and the respective treatment duration is determined by the patient path. We focus on probabilistic patient paths with stochastic routing between treatment operations and random treatment durations. In the CHE case, we distinguish four types of patients with different paths (type I - IV). Type I patients are cardiothoracic surgical patients who, after surgery, are admitted to a special postoperative care unit and are expected to return to the ward on the following day. There is a 15% chance that complications require an admission to the Intensive Care unit (ICU). Type II patients follow a fast-track variant of the type I path such that patients are expected to return to the ward on the day of surgery. Concerning type III and IV patients, we focus on their possible

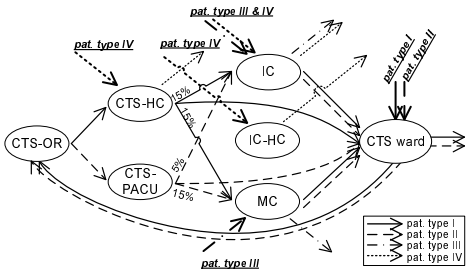


Figure 1: Interference of type I - IV patient pathways at the different hospital units

interference with type I and II patients at the ICU. The pathways are depicted in Figure 1. Here, the circles represent the hospital units involved and arrows represent possible adjacent treatment operations.

3 Experiments

The settings of our simulation experiments are based on the case analysis at the cardiothoracic surgery department of the CHE. In Table 1 the simulation outcomes for the basic setup based on the current situation at the CHE are shown. Outcome measures are the patient throughput, i.e. the number of patients discharged from the hospital after treatment, and the costs for regular and back-up bed capacity. At the CHE, an annual throughput of about 1800 type I and II patients is achieved. Thus, the performance of the agent-based simulation system compares well to the human planners. Regarding admission requests for type III and IV patients, the system achieves an acceptance rate of about 82.93% and 98.97%, respectively. These outcomes are comparable to recent aggregated measurements performed at the CHE.

In many hospitals, an efficient allocation of resources to the different hospital units is a major managerial

Outcome measure	Mean$\pm$Stdev
Type I + II patient throughput	1768.08 ± 40.31
Type III patient throughput	539.16 ± 26.91
Type IV patient throughput	899.72 ± 10.28
Resource costs	
regular	38835 ± 0
back-up	355.65 ± 48.64

Table 1: Simulation outcomes for basic scenario

issue because the relationship between beds, occupancy and acceptation rates for different patient groups is complex [2]. Several allocation policies were assessed in scenario analyses. Furthermore, we implemented a brute-force optimizer that uses the simulation system to optimize the bed allocations. In the full paper, first optimization results are presented using the mean costs per patient as objective function.

4 Conclusions

We presented an agent-based simulation and evaluation tool for patient admission scheduling that realistically captures the complex features of the problem domain. To the best of our knowledge, this is the first agent-based simulation system for patient admission scheduling that includes multiple patient groups with stochastic arrival and treatment pathways. The implemented simulation system can be adjusted to comparable situations in other hospital settings. Furthermore, extensive simulation experiments demonstrate the applicability of the model and show how the agent-based simulation tool is useful for decision support.

References

[1] L. Maruster et al. Logistic-based patient grouping for multi-disciplinary treatment. *Artificial Intelligence in Medicine*, 26:87–107, 2002.

[2] P.R. Harper and A.K. Shahani. Modelling for the planning and management of bed capacities in hospitals. *Journal of the Operational Research Society*, 53:11–18, 2002.

An Empirical Study of Instance-based Ontology Matching

Antoine Isaac [a,b] Lourens van der Meij [a,b] Stefan Schlobach [a]

Shenghui Wang [a,b]

[a] *Vrije Universiteit Amsterdam*
[b] *Koninklijke Bibliotheek, Den Haag*

Abstract

Instance-based ontology mapping is a promising family of solutions to a class of ontology alignment problems. It crucially depends on measuring the similarity between sets of annotated instances. In this paper we study how the choice of co-occurrence measures affects the performance of instance-based mapping. To this end, we have implemented a number of different statistical co-occurrence measures. We have prepared an extensive test case using vocabularies of thousands of terms, millions of instances, and hundreds of thousands of joint items. We have obtained a human Gold Standard judgement for part of the mapping-space. We then study how the different co-occurrence measures and a number of algorithmic variations perform on our benchmark dataset as compared against the Gold Standard. Our systematic study shows excellent results of instance-based matching in general, where the more simple measures often outperform more sophisticated statistical measures.

This paper is an abbreviated version of a paper accepted at the 6th International Semantic Web Conference, ISWC 2007 [3].

1 Introduction

The semantic heterogeneity problem is probably the single-most urgent problem to be solved to realise a web-scale Semantic Web. This makes automatic ontology mapping – determining relations such as equivalence or subsumption between concepts of two separate ontologies – as the anticipated solution to semantic heterogeneity, a research issue of paramount importance [1].

In this paper we focus on instance-based ontology mapping, *i.e.* the construction of links between concepts based on their instances. In instance-based mapping semantic relations between concepts of two ontologies are determined based on the overlap of their instance sets. The idea for mapping is then simply that the higher the ratio of co-occurring instances for two concepts, the more related they are. The difficult question is how to define the notion of significance for such *extension* overlap. We propose a systematic approach considering the following dimensions:

- **Measures:** the most simple idea is to calculate the common factor of two concepts C and D, for example, the Jaccard measure which measures the proportion of jointly annotated books over all books annotated by C and D individually. In statistics and Information Theory a number of other measures have also been developed, such as Pointwise Mutual Information, Information Gain or Log-likelihood ratio.
- **Thresholds:** often the above mentioned measures are vulnerable for data-sparseness: if there are too few instances, the common factor measure ranks mappings high when the two concepts involved are each only used once to annotate the same book. The solution to dealing with this issue is to consider thresholds in the measures.
- **Hierarchy:** following the semantics of ontologies we can use the hierarchy, *i.e.* including the instances of descendants in the extension of a concept.

Based on a case where two book collections indexed with different thesauri overlap, we answer the following research questions:

1. Is instance-based mapping a reliable technology to be applied in practical, possibly critical applications?
2. Which combination of measures, thresholds and information inclusion works best, possibly depending on circumstances such as whether precision or recall is considered more important?

2 Use case

Our case is situated at the National Library of the Netherlands, which maintains a large number of collections. Two of them are the *Deposit Collection*, containing all the Dutch printed publications, and the *Scientific Collection*, mainly about the history, language and culture of the Netherlands. Each collection is described according to its own indexing system: The Scientific Collection uses the *GTT thesaurus* – 35,000 concepts – while the books contained in the Deposit Collection are indexed against the *Brinkman thesaurus* – 5,000 concepts. Both thesauri have similar coverage but differ in granularity. Around 250,000 books are common to the depot and scientific collections, and have therefore been manually annotated with both GTT and Brinkman vocabularies.

3 Experimental Setup

We have run experiments with the Jaccard, Pointwise Mutual Information, Information Gain and Log-likelihood ratio measures, as well as with an version of Jaccard corrected to compensate concepts with very few joint instances. For each measure we calculate four ordered lists, two taking the hierarchy into account, two not, of which one is with a threshold of 1, and one with a threshold of 10.

To evaluate these results, we have developed a procedure consisting of three steps: *producing a Gold Standard* based on sample data (1.600 mappings), calculating *average precision* over ranked mappings and *approximating recall* using a set of mapping obtained by lexical similarity between concepts' labels.

4 Results of the experiments

Generally, the results are surprisingly good, as compared to results from other ontology matching evaluations on the same data [2]. This indicates that instance-based matching would probably be an easier task than structure-based or label-based mapping methods, when there are instances available. This also indicates that our technique will be suitable in critical applications. Based on our experiments, we learn the following lessons:

- including a threshold generally improves precision but there is a price to pay in terms of the recall;
- considering hierarchical information to calculate concepts' extension does no bring significant improvement, and in most cases even decreases performances;
- in our case, the most simple measures Jaccard and its corrected version perform the best.

References

[1] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.

[2] Jérome Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. ISWC Ontology Matching workshop, 2007.

[3] Antoine Isaac, Lourens van der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance-based ontology matching. In *Proceedings of the the 6th International Semantic Web Conference (ISWC 2007)*, Busan, Korea, 2007.

The Importance of Link Evidence in Wikipedia

Jaap Kamps [a,b] Marijn Koolen [a]

[a] *Archives and Information Studies, University of Amsterdam, The Netherlands*
[b] *ISLA, University of Amsterdam, The Netherlands*

Abstract

The full paper appeared as: Jaap Kamps and Marijn Koolen. The importance of link evidence in Wikipedia. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Rutven, and Ryen W. White, editors, *Advances in Information Retrieval: 30th European Conference on IR Research (ECIR 2008)*, volume 4956 of *Lecture Notes in Computer Science*, pages 270–282. Springer Verlag, Heidelberg, 2008.

1 Introduction

Wikipedia is one of the most popular information sources on the Web. The free encyclopedia is densely linked. The link structure in Wikipedia differs from the Web at large: internal links in Wikipedia are typically based on words naturally occurring in a page, and link to another semantically related entry. Our main aim is to find out if Wikipedia's link structure can be exploited to improve ad hoc information retrieval. We first analyse the relation between Wikipedia links and the relevance of pages. We then experiment with use of link evidence in the focused retrieval of Wikipedia content, based on the test collection of INEX 2006.

2 Link Evidence in Wikipedia

2.1 Analysis

We have conducted an extensive analysis of Wikipedia link structure. Figure 1 shows the global link indegree

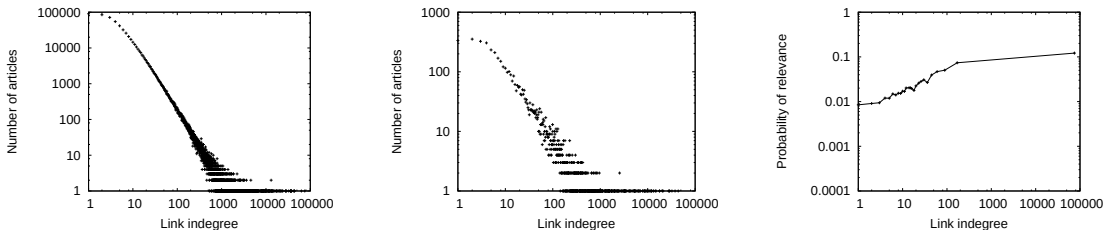


Figure 1: Wikipedia link degree distribution off all pages (left), of relevant pages (middle) and prior probability of relevance (right)

distribution of all pages (left), and of all pages relevant for an INEX 2006 topic (middle), allowing us to calculate the prior probability of a page being relevant given its indegree (right). We clearly see an increasing curve, suggesting that link evidence can be used as (possibly weak) indicator of relevance. For retrieval we have to combine content-based retrieval scores with a score based on the link topology. The crucial issue when incorporating link evidence is to retain the focus on the topic at hand, and avoid the retrieval of important but off-topic pages.

2.2 Priors

On top of a standard language model for information retrieval, we implemented a range of link evidence priors making the link evidence increasingly sensitive to the local context. First, we use a *standard indegree*

prior by multiplying the retrieval score with 1+ the indegree:

$$P_{\text{standard}}(d) \propto 1 + indegree(d).$$

Here, the indegree score for a page may be based on either the *global* link graph or the *local* link graph restricted to pages with a content-based retrieval score. Second, we use a *log indegree prior* using the logarithm of the indegrees:

$$P_{\log}(d) \propto 1 + log(1 + indegree(d)).$$

The logged indegree values will reduce the impact of the indegrees and hence may act as a safe-guard against the infiltration of loosely related pages with very high (global) indegrees. Again, the indegree score may be based on *global* or *local* evidence.

We experiment with weighting the local indegree (the number of links from pages in the relevant set) by its global indegree (the number of links from arbitrary pages). That is, our third prior is a combination of the local *and* global link evidence computed as:

$$P_{\text{LocGlob}}(d) \propto 1 + \frac{indegree_{\mathsf{local}}(d)}{1 + indegree_{\mathsf{global}}(d)}.$$

This is similar to the well-known *tf.idf* weighting scheme used to determine term importance.

2.3 Results

The scores for three INEX 2006 ad hoc retrieval tasks are in Table 1. We see that the global link evidence

Table 1: Results of link evidence on three INEX 2006 ad hoc retrieval tasks. Best scores are in bold-face. Significance levels are 0.05 (*), 0.01 (**), and 0.001 (***).

Run ID	Thorough		Focused		Relevant in Context	
	MAep,off		nxCG@10,off		MAgP	
Baseline	0.0353		0.3364		0.1545	
Global Indegree	0.0267	-24.40***	0.1979	-41.16***	0.1073	-30.57***
Log Global Indegree	0.0335	-4.99	0.3066	-8.87**	0.1352	-12.50***
Local Indegree	0.0405	+14.75*	0.3218	-4.34	0.1467	-5.02*
Log Local Indegree	0.0418	+18.46***	0.3460	+2.85	0.1515	-1.96
Local/Global Indegree	**0.0463**	+31.08***	**0.3629**	+7.88**	**0.1576**	+1.99*

leads to a loss of performance, it tends to favor important but off-topic pages. The more conservative local link evidence fares much better, with the less aggressive logged version leading to improvement on two of the tasks and a small loss on the third. The even more careful combined local/global indegree prior is effective on all tasks.

3 Findings

Our main findings are: First, our analysis of the link structure reveals that the Wikipedia link structure is a (possibly weak) indicator of relevance. Second, our experiments on INEX ad hoc retrieval tasks reveal that if the link evidence is made sensitive to the local context we see a significant improvement of retrieval effectiveness. Hence, in contrast with earlier TREC experiments using crawled Web data, we have shown that Wikipedia's link structure can help improve the effectiveness of ad hoc retrieval.

Acknowledgments This research was supported by the Netherlands Organization for Scientific Research (NWO, grants # 639.072.601, 612.066.513, and 640.001.501), and the E.U.'s 6th FP for RTD - project MultiMATCH contract IST-033104.

References

[1] Jaap Kamps and Marijn Koolen. The importance of link evidence in Wikipedia. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Rutven, and Ryen W. White, editors, *Advances in Information Retrieval: 30th European Conference on IR Research (ECIR 2008)*, volume 4956 of *Lecture Notes in Computer Science*, pages 270–282. Springer Verlag, Heidelberg, 2008.

Evolutionary Dynamics for Designing Multi-Period Auctions[1]

Tomas Klos Gerrit Jan van Ahee

Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, The Netherlands

Mechanism design (MD) has recently become a very popular approach in the design of distributed systems of autonomous agents. Also called 'inverse game theory' [4], MD is concerned with designing the games or systems in which agents interact, and to do this in such a way that rational agent behavior in those games leads to certain desirable properties for the system as a whole. A key assumption in MD is that *agents behave rationally*, since this provides the predictability of agent behavior required for optimizing the mechanism's design. In many practical circumstances, however, agents don't behave rationally since, in general, finding Nash equilibrium strategies to play is intractable [2].

Because of such negative results, many have resorted to heuristic approaches to these problems. Here, we propose studying the interaction between the mechanism designer and the game participants as a higher level, 'meta-game,' in which the designer chooses among alternative mechanism designs, while the agents choose among alternative strategies to play. We solve this game 'heuristically' using evolutionary game theory techniques, specifically, the (coupled) replicator dynamics (RD) [3]. To illustrate, we adopt the multi-period auction scenario developed by Pardoe and Stone (PS) [5].

1 Analyzing Meta-Games

Pardoe and Stone [5] studied a setting in which a seller wants to sell a large number of identical goods, and chooses to do so by repeatedly auctioning off batches of 60 items in a sequence of uniform-price sealed-bid auctions. In each auction (of 60 goods), all winning bidders pay the price of the highest losing bid. There are 120 bidders for each auction, and each receives a 'signal,' indicating the value of each of the (identical) goods, drawn uniformly at random from $[0, 1]$. Bidders are not assumed to bid strictly rationally, but to probabilistically choose one of a limited set of 5 'heuristic' bidding strategies (which we don't detail here, see [5]), $\{$EQ, OE, UE, DO, AV$\}$. Given such bidders, the seller in turn, is unsure whether it is most profitable to auction off all 60 items at once, or to distribute them evenly over $2, 3$, or 4 'periods,' thereby revealing the winning price in between periods, and allowing the remaining bidders to update their bids.

Whereas PS need to perform extensive simulation experiments, we have designed an efficient algorithm which allows us to exactly calculate (1) expected revenues for the seller's choices of 'number of periods,' and (2) the bidders' utilities of using each of the 5 heuristic strategies. This algorithm allows us to study the interaction between the seller and the bidders as a meta-game, which we do using coupled replicator equations [1, 3, 6, 7] for the two sets of strategies. The RD are a popular and intuitive way of modeling deterministic evolutionary dynamics in games [3]. With RD, the state of a population is represented as a vector of relative frequencies of different strategies. In our case, there are two 'populations' of strategies, viz. the seller's vector $\mathbf{x} = (x_1, \ldots, x_n)$ with $n = 4$ choices for number of periods, and the bidders' vector $\mathbf{y} = (y_1, \ldots, y_m)$ with $m = 5$ choices of heuristic strategies for the bidders. In each of a sequence of 'generations,' the states of these populations are evolved using replicator equations, which express that above (below) average strategies become more (less) prevalent in their population:

$$\frac{\dot{x}_i}{x_i} = ((\mathbf{Ay})_i - \mathbf{x} \cdot \mathbf{Ay}) \qquad \text{and} \qquad \frac{\dot{y}_j}{y_j} = ((\mathbf{Bx})_j - \mathbf{y} \cdot \mathbf{Bx}).$$

Here, x_i and y_i are the densities of pure strategies for the seller and the bidders, respectively, a dot indicates derivative wrt time, $\mathbf{A}$ is the payoff (revenue) matrix of the seller, and $\mathbf{B}$ the payoff (utility) matrix of the

[1]Published in: Proceedings Autonomous Agents and Multi-Agent Systems (AAMAS), p. 1589–1592, IFAAMAS, 2008.

bidders. With *coupled* RD, the growth rate for each strategy in each population depends on the distribution of strategies in the other population.

2 Experiments

We performed experiments in which the seller's and the bidders' strategy distributions are evolved using RD based on the strategies' expected average revenues and utilities, respectively. Figure 1 shows an example where the initial bidding strategy distribution is drawn from the 4-simplex uniformly at random. The seller's initial probability distribution is always uniform.

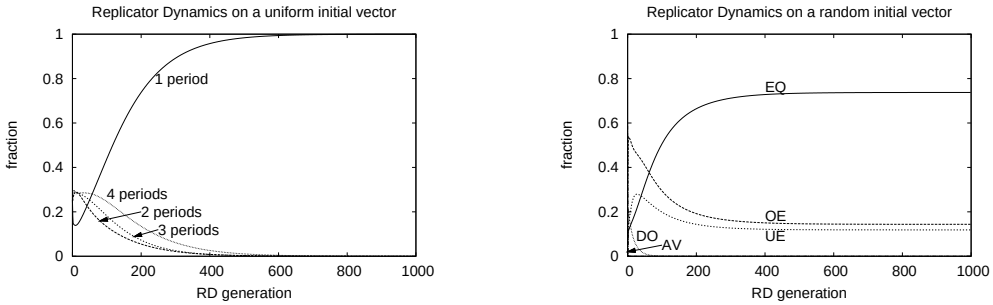


Figure 1: Coupled replicator dynamics.

In Figure 2 (left) we plot the number of samples in which each of the numbers of periods maximizes revenue—before and after RD. Choosing 1 or 2 periods generates the highest performance in 82% of all

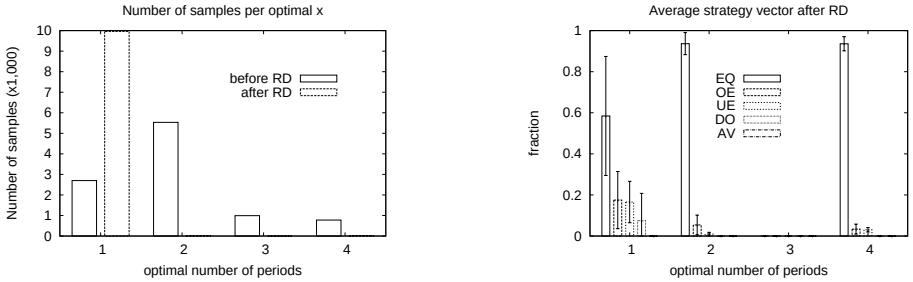


Figure 2: Vectors after RD.

randomly generated distributions (before RD). After RD, choosing 1 period is optimal virtually always: 2 (4) periods is optimal in only 19 (21) out of 10,000 samples. The graph on the right shows the average strategy distributions occuring in those cases, showing a high prominence of the EQ bidding strategy, although when 1 period is optimal, the high standard deviation (errorbar) suggests maybe we're averaging over several qualitatively distinct attractors. Further research is in order here.

References

[1] T. Börgers and R. Sarin. Learning through reinforcement and replicator dynamics. *J. Ec. Th.*, 77, 1997.

[2] L.-C. Chen and X. Deng. Settling the complexity of 2-player Nash equilibrium. In *FOCS*. IEEE, 2006.

[3] J. Hofbauer and K. Sigmund. Evolutionary game dynamics. *Bulletin of the AMS*, 40, 2003.

[4] C. H. Papadimitriou. Algorithms, games and the internet. In *STOC*. ACM, 2001.

[5] D. Pardoe and P. Stone. Developing adaptive auction mechanisms. *ACM SIGecom Exchanges*, 5, 2005.

[6] Y. Sato and J. P. Crutchfield. Coupled replicator equations for the dynamics of learning in multiagent systems. *Phys. Rev. E*, 67, 2003.

[7] K. Tuyls, A. Nowé, T. Lenaerts, and B. Manderick. An evolutionary game theoretic perspective on learning in multi-agent systems. *Synthese*, 139, 2004.

Combining Expert Advice Efficiently

Wouter M. Koolen[a] Steven de Rooij[b]

[a] *Centrum voor Wiskunde en Informatica, P.O.Box 94079 1090 GB Amsterdam, NL*
[b] *Technische Universiteit Eindhoven, P.O.Box 513 5600 MB Eindhoven, NL*

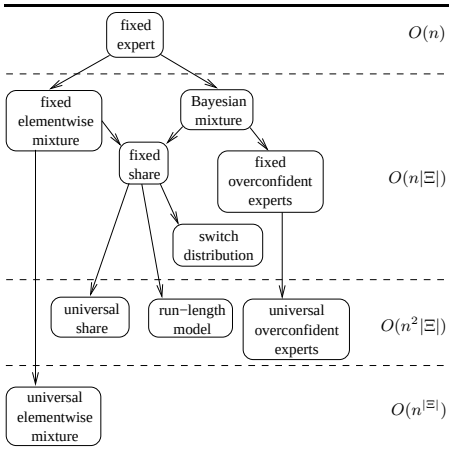
The original paper appears in the Proceedings of the 21st Conference on Computational Learning Theory (COLT 2008). A longer, but unrefereed version is available on the CORR Computer Science arXiv as arXiv:0802.2015.

Abstract

We show how models for prediction with expert advice can be defined concisely and clearly using hidden Markov models (HMMs); standard HMM algorithms can then be used to efficiently calculate how the expert predictions should be weighted according to the model. We cast many existing models as HMMs and recover the best known running times in each case. We also describe two new models: the switch distribution, which was recently developed to improve Bayesian/Minimum Description Length model selection, and a new generalisation of the fixed share algorithm based on run-length coding. We give loss bounds for all models and shed new light on the relationships between them.

Introduction We cannot predict exactly how complicated processes such as the weather, the stock market, social interactions and so on, will develop in the future. Nevertheless, people do make weather forecasts and buy shares all the time. Such predictions can be based on formal models, or on human expertise or intuition. An investment company may even want to choose between portfolios on the basis of a combination of these kinds of predictors. In such scenarios, predictors typically cannot be considered "true". Thus, we may well end up in a position where we have a whole collection of prediction strategies, or *experts*, each of whom has *some* insight into *some* aspects of the process of interest. We address the question of how a given set of experts can be combined into a single prediction strategy that is as good as, or if possible even better than, the best individual expert, or combination of experts.

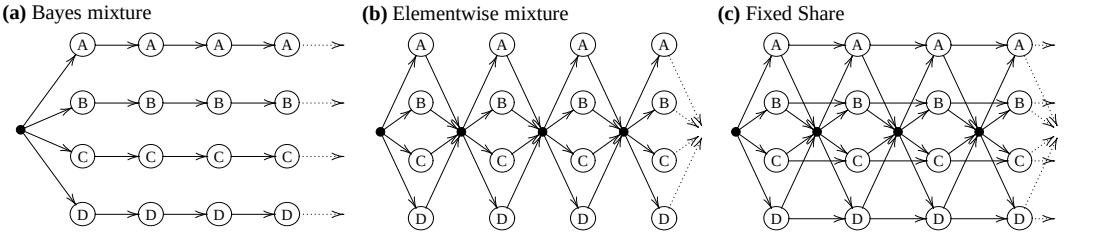
Figure 1: Prediction schemes: generalisation graph with run times



Setup Let Ξ be a finite set of experts. Each time step, each expert $\xi \in \Xi$ issues a distribution $P_\xi(\boldsymbol{x}_{n+1}|x^n)$ on the next outcome $\boldsymbol{x}_{n+1}$ given the previous observations $x^n := x_1, \ldots, x_n$. Here, each outcome x_i is an element of some countable space $\mathcal{X}$, and random variables are written in bold face. The probability that expert ξ assigns to a sequence of outcomes is given by the chain rule: $P_\xi(x^n) = P_\xi(x_1) \cdot P_\xi(x_2|x_1) \cdot \ldots \cdot P_\xi(x_n|x^{n-1})$.

Bayes Mixture The traditional Bayesian approach to combine the expert predictions is to conceive a prior w on the experts Ξ, and form the joint distribution $P_w(x^n, \xi) := w(\xi)P_\xi(x^n)$. Inference is then based on this joint distribution. Given observations x^n, we can compute the *marginal probability* of the data $P_w(x^n) = \sum_{\xi\in\Xi} P_w(x^n, \xi)$, as well as the *posterior distribution* on the experts $P_w(\boldsymbol{\xi}|x^n) = P_w(x^n, \boldsymbol{\xi})/P_w(x^n)$, and the *predictive distribution* on the next outcome $P_w(\boldsymbol{x}_{n+1}|x^n) = \sum_{\xi\in\Xi} P_w(\xi|x^n)P_\xi(\boldsymbol{x}_{n+1}|x^n)$. Thus the Bayesian prediction strategy sequentially predicts the outcomes by weighing the experts' predictions according to the posterior. This simple probabilistic approach has the advantage that it is computationally

Figure 2 Existing models rendered as Hidden Markov Models



easy: predicting n outcomes using $|\Xi|$ experts requires only $O(n \cdot |\Xi|)$ time. Additionally, this Bayesian strategy guarantees that the overall probability of the data is only a factor $w(\hat{\xi})$ smaller than the probability of the data according to the best available expert $\hat{\xi}$. On the other hand, with this strategy we never do any *better* than $\hat{\xi}$ either: we have $P_{\hat{\xi}}(x^n) \geq P_w(x^n) \geq P_{\hat{\xi}}(x^n)w(\hat{\xi})$, which means that potentially valuable insights from the other experts are not used to our advantage!

ES-Priors The Bayesian strategy performs well if there is a single best expert, but it may be outperformed by more sophisticated combinations of experts, that, for example, mix the experts' predictions, or switch between experts over time. We introduce a new, general way to model such combinations using *expert sequence priors*. We conceive a prior π on sequences of experts, and form the joint $P_\pi(x^n, \xi^n) := \pi(\xi^n)P_{\xi^n}(x^n)$, where, for a fixed sequence of experts ξ^n, the prediction strategy P_{ξ^n} predicts using expert ξ_i at time i. Thus, $P_{\xi^n}(x^n) = P_{\xi_1}(x_1)P_{\xi_2}(x_2|x_1)\cdots P_{\xi_n}(x_n|x^{n-1})$. In the sequential perspective, after observations x^n, the ES strategy mixes the experts' predictions of the next outcome according to the posterior distribution on the next expert: $P_\pi(\boldsymbol{x}_{n+1}|x^n) = \sum_{\xi_{n+1}\in\Xi} P_\pi(\xi_{n+1}|x^n)P_{\xi_{n+1}}(\boldsymbol{x}_{n+1}|x^n)$. The ES strategy performs as well as the best expert sequence $\hat{\xi^n}$ for the data x^n, modulo the prior: $P_{\hat{\xi^n}}(x^n) \geq P_\pi(x^n) \geq P_{\hat{\xi^n}}(x^n)\pi(\hat{\xi^n})$. Traditional Bayes with prior w is recovered by choosing $\pi(\xi, \xi, \xi, \ldots) = w(\xi)$ for each expert $\xi \in \Xi$. One may craft more interesting ES priors, with better performance guarantees at modest overheads. In the full paper, we give the ES priors corresponding to the schemes for prediction with expert advice that are listed in Figure 1. It includes several important schemes from the literature, and the new *Switch Distribution* and *Run-length model*.

HMMs To compute the ES posterior on the next expert $P_\pi(\boldsymbol{\xi}_{n+1}|x^n)$, we must sum the joint over all expert pasts ξ^n. In general, this requires time exponential in n, so not all ES-priors lead to efficient prediction schemes. However, for the prediction schemes listed in Figure 1, the ES-prior can be specified as the marginal distribution of a Hidden Markov Model. We have shown the HMMs for three simple prediction schemes in Figure 2. The more advanced schemes that we consider in the paper require infinite, non-stationary HMMs. Despite this expressive power, these HMMs have modest size. This is key to time efficiency, as the size is directly related to the running time of the Forward Algorithm, which is used for sequential prediction.

Conclusion We introduced ES-priors, which unify description of many existing prediction strategies with expert advice from the literature including (Bayesian) statistics, source coding and universal prediction. Hidden Markov Models can be used as intuitive graphical descriptions of ES-priors, and moreover small HMMs correspond to computationally efficient prediction strategies. We use this new approach to describe and analyse several important existing models, as well as two new models for expert tracking.

Paying Attention to Symmetry

Gert Kootstra Arco Nederveen Bart de Boer

[a] *University of Groningen, P.O.Box 407, 9700 AK, Groningen*

Abstract

Humans are very sensitive to symmetry in visual patterns. Symmetry is detected and recognized very rapidly, and eye fixations are concentrated along the axis of symmetry or the symmetrical center of the patterns. This suggests that symmetry is a highly salient feature. Existing computational models of saliency, however, have mainly focused on contrast as a measure of saliency. These models do not take symmetry into account. In this paper, we discuss local symmetry as a measure of saliency. We developed a number of symmetry models and performed an eye-tracking study with human participants viewing photographic images to test the models. The results show that the symmetry models better match the human data than the contrast saliency model of Itti, Koch and Niebur [1]. This indicates that symmetry is a salient structural feature for humans, a finding which can be exploited in computer vision.

1 Introduction

Symmetry is a prominent visual feature in our daily environments. Symmetrical forms have a high esthetic value. Faces with enhanced symmetry, for instance, are judged more attractive than the original faces [2]. It is also known that the human visual system is sensitive to symmetry. Symmetrical forms, especially when possessing multiple symmetry axes, are detected very rapidly [3]. Also recall and discrimination performances increase with symmetrical patterns [4]. Furthermore, humans interpret symmetrical regions as figure, and asymmetrical regions as background [5].

This sensitivity to symmetry suggests that symmetry is something that humans pay attention to, and can therefore be considered as a salient feature. If we look at the second column of figure 1, we indeed see that humans clearly have a preference to fixate on the centers of the symmetrical forms. Most existing models of saliency, however, are based on contrast (e.g., [1, 6]). Although these models are able to predict human eye fixations to some extend [6-8], they do not predict the attention to symmetry in figure 1 (third column).

This apparent deficiency in current vision models was the motivation for the present study. In this paper, we investigate the role of local symmetry in visual attention. We extended a number of symmetry operators to multi-scale saliency models. Furthermore, we performed an eye-tracking study to test the models, and compared the performance with the contrast saliency model of Itti *et al.*[1].

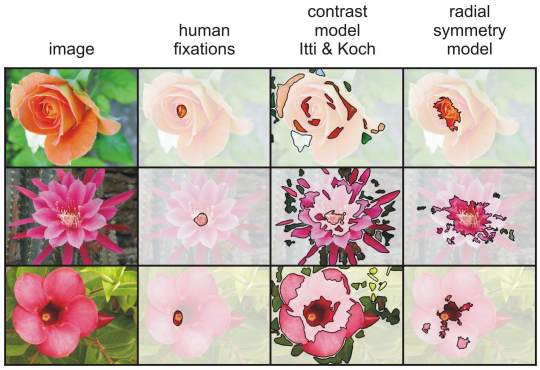


Figure 1: Examples of images containing symmetrical forms. The second column shows the human fixation density map, the third shows the contrast saliency map, and the last shows our symmetry saliency map. The bright regions are the parts of the maps above 50% of its maximum. The preference of humans to fixate on the center of symmetry is correctly reproduced by our symmetry model, whereas the contrast model displays a wide non-specific saliency response.

2 Methods

The symmetry models are based on symmetry operators proposed in [9, 10]. The basis of all models is the same: A symmetry kernel is applied to every pixel in the image. In the kernel, the presence of symmetry is checked for all possible symmetry axes. Symmetry is present if the gradients in the kernel are mirror symmetric with respect to the symmetry axes. These symmetry calculations are done on five different scales, to achieve scale invariance. The

isotropic symmetry model is as described above. The *radial symmetry model* responds stronger to patterns containing multiple symmetry axes, and the *color symmetry model* uses color instead of luminance information.

To test the saliency model, we performed an eye-tracking experiment. 31 participants where shown 100 photographic images, and their eye gaze was recorded. The images were from five cateogries: images containing natural symmetries, animals, street scenes, buildings, and nature scenes.

3 Results

The resulting saliency maps of the symmetry models are compared with the human data by calculating the correlation coefficient. The results show significantly higher correlation for the symmetry models than for the contrast model of Itti and Koch. The performance of the symmetry models is similar to the inter-participant correlation, showing that the models predict the participant's eye fixations similarly good as the fixations of other participants would.

4 Discussion

Our results show a significantly higher correlation for the symmetry saliency models than for the contrast saliency model. This suggests that humans pay attention to symmetry, and that symmetry can be considered a salient feature.

Our findings have potentially interesting applications in artificial vision systems. Specifically robots navigating in man-made environments containing many symmetrical patterns could benefit from our symmetry models to select interesting and useful visual information.

References

[1] L. Itti, C. Koch, and E. Niebur, A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254-1259, 1998.

[2] K. Grammer and R. Thornhill, Human (Home sapiens) Facial Attractiveness and Sexual Selection: The Role of Symmetry and Averageness, *Journal of Comparative Psychology*, vol. 108, pp. 233-242, 1994.

[3] S. E. Palmer and K. Hemenway, Orientation and Symmetry: Effects of Multiple, Rotational, and Near Symmetries, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 4, pp. 691-702, 1978.

[4] F. L. Royer, Detection of Symmetry, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 7, pp. 1186-1210, 1981.

[5] J. Driver, G. C. Baylis, and R. D. Rafal, Preserved figure-ground segregation and symmetry perception in visual neglect, *Nature*, vol. 360, pp. 73-75, 1992.

[6] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, A Coherent Computational Approach to Model Bottom-Up Visual Attention, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 802-817, 2006.

[7] D. J. Parkhurst, K. Law, and E. Niebur, Modeling the Role of Salience in the Allocation of Overt Visual Attention, *Vision Research*, vol. 42, pp. 107-123, 2002.

[8] N. Ouerhani, R. von Wartburg, H. Hügli, and R. Müri, Empirical Validation of the Saliency-based Model of Visual Attention, *Electronic Letters on Computer Vision and Image Analysis*, vol. 3, pp. 13-14, 2004.

[9] D. Reisfeld, H. Wolfson, and Y. Yeshurun, Context-Free Attentional Operators: The Generalized Symmetry Transform, *International Journal of Computer Vision*, vol. 14, pp. 119-130, 1995.

[10] G. Heidemann, Focus-of-Attention from Local Color Symmetries, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 817-830, 2004.

Of Mechanism Design and Multiagent Planning[1]

Roman van der Krogt [a] Mathijs de Weerdt [b] Yingqian Zhang [b]

[a] *Cork Constraint Computation Centre, University College Cork, Ireland*
[b] *Delft University of Technology, Delft, The Netherlands*

Many interesting applications of AI planning feature an environment with multiple agents. Often these agents represent companies or other autonomous entities which may have (partially) conflicting preferences. Such self-interested agents may be tempted to lie about their costs or the actions they can do in order to obtain an outcome that is more rewarding for them. We therefore study the multiagent planning problem from a mechanism design perspective, showing how to incentivise agents to be truthful. Below we first introduce our model of multiagent planning problems for self-interested agents, then we analyse where known results in mechanism design fail to deal with multiagent planning, and we propose a solution to this problem.

Formally, a *multiagent planning problem* $\theta \in \Theta$ for a set of agents is a tuple $\theta = (\theta_1, \ldots, \theta_n)$ of private planning problems $\theta_i \in \Theta_i$ for these agents. Agent i's planning problem θ_i consists of *(i)* a set of ground atomic formulae; *(ii)* a set of actions this agent may carry out; *(iii)* a cost function c_i that assigns a cost to each operator; *(iv)* that part of the (common) initial state the agent is aware of; *(v)* a set of goals G_i; and *(vi)* a reward function $r_i : G_i \rightarrow \mathbb{R}^+$, assigning a reward to each of the goals. The goals of different agents *can* be mutually exclusive. The solution to a multiagent planning problem is a plan π, which is a partially ordered sequence of actions. The space of all plans is denoted by Π. The *utility* of plan π is defined as: $U(\pi, \theta) = c(\pi, \theta) + r(\pi, \theta)$, where $c(\pi, \theta)$ is the cost of executing the plan and $r(\pi, \theta)$ denotes the *revenue* of π that is given by the reward functions for the goals that have been attained. An optimal planner returns the plan which has the highest utility.

Each agent i has preferences over the possible plans defined by its *valuation* $v_i(\pi, \theta) = c_i(\pi, \theta) + r_i(\pi, \theta)$. In this paper, we consider a mechanism design problem where the declaration of the type of all agents is the input, and a plan $\pi \in \Pi$ is the output of the mechanism. A mechanism using an optimal planning algorithm will choose the best plan in Π, which maximises the *social welfare* $v(\pi, \theta)$ that is the total valuation of the agents. The social welfare can be maximised only if the agents report their types truthfully. In order to achieve this, *payments* are introduced to penalise some agents and possibly reimburse some others based on their contribution to the social welfare. With payments, the *utility* of the agent i on the outcome π is defined by: $u_i(\pi, \theta) = v_i(\pi, \theta) - p_i(\theta)$. This utility is what rational agents aim to maximise.

We consider a mechanism to be a tuple $(f, p_1, \ldots, p_n)$ where $f : \Theta_1 \times \cdots \times \Theta_n \rightarrow \Pi$ is a planning function, and $p_1, \ldots, p_n$ are payment functions which specify for each agent the mount it pays. The goal of mechanism design for MAP is thus to find a mechanism $(f, p_1, \ldots p_n)$ such that $f(\theta)$ returns the plan which maximises the social welfare. We say a mechanism is *truthful* iff no agent can achieve a higher utility by lying about its type.

(Deposit-)VCG Mechanisms for MAP

When agents declare their type, they can lie in three different ways: *(i)* about the *value* of a plan, i.e. the costs, the rewards, and the goals; *(ii) under-reporting* the available actions; and *(iii) over-reporting* non-existing actions or states. We investigate how to design the truthful mechanisms to prevent such lying types for MAP. So-called Vickrey-Clarke-Groves (VCG) mechanisms are very successful in satisfying this property [1].

Previous work has shown that every VCG mechanism is truthful [1]. Indeed, we show that *the VCG mechanism for MAP with an optimal algorithm prevents lying about values and under-reporting, or a combination of both*. Using an optimal planning algorithm, VCG mechanisms work that well, because *(i)* the agents' utility and thus their incentives are aligned with the social welfare, and moreover *(ii)* the goal of the

[1]The full paper appeared in the Proc. of the 18th European Conference on Artificial Intelligence (ECAI'08), pp. 423-427, IOS Press.

algorithm is also to maximise the social welfare. Surprisingly, for the third type of lying, i.e. over-reporting, an agent *can* gain from reporting *more* than it has at its disposal. *The VCG mechanism with an optimal algorithm for MAP cannot prevent over-reporting.* Intuitively, the reason for this is that the outcome of MAP is a global, distributed plan that only achieves its value upon *successful* execution. This gives the agents additional ways to cheat which aren't prevented by the VCG mechanism: their penalties are based on what they *promise* to do; not on what they *actually* achieve. An agent may for example be rewarded for actions that it claims it has and that help other agents to achieve their goals, but which it cannot actually execute. If those actions are included in the generated plan π, the utility of π is not representing the social welfare. So even an optimal planner cannot guarantee to output a "best" plan which maximises the social welfare.

In order to avoid over-reporting, we introduce the *deposit-VCG* mechanism: first, the mechanism asks the agents to declare their types θ_i, then it asks each agent to pay the amount $r(G)$ (the total award of the goals in G) as a deposit. The mechanism then finds a plan π using an optimal algorithm f, taking into account only the agents who paid the deposit. After each agent i pays p_i according to the VCG formula, the mechanism informs the agents of the plan π, and each agent i executes its part. If any local plan fails due to the agent i's declaration, agent i will not get its deposit back. All other agents are returned their deposits. Since the separate deposit stage does not enlarge the strategy space of the agents, it is straightforward to see that if the agents are truthful under the VCG mechanism, they will not be better off by lying under the deposit-VCG mechanism. Consequently, deposit-VCG is truthful with respect to lying about values and under-reporting. Moreover, it also prevents over-reporting. Thus, *the deposit-VCG mechanism with an optimal algorithm is truthful for MAP*.

(Deposit-)VCG-based Approximations for MAP

The (deposit-)VCG mechanism requires that f makes optimal decisions. Except for some specific domains, this is intractable, as planning in general is PSPACE-complete. Hence, it is desirable to develop a *truthful*, *polynomial-time* mechanism which can produce reasonable results. We will call a mechanism *deposit-VCG-based*, if f is a sub-optimal algorithm and $p(\cdot)$ is calculated according to the deposit-VCG mechanism. Unfortunately, deposit-VCG-based mechanisms are generally not truthful. The reason is that VCG payments align the agent's utility with the value of the system's solution. Therefore by lying, an agent may "help" a non-optimal mechanism to achieve a better solution, and thus make more profit for itself.

It has been shown in [2] that a mechanism is truthful if the algorithm f is *maximal in its range* (MIR). Informally speaking, a planning algorithm f is MIR if it optimises the social welfare by selecting the best plan out of an on *forehand determined* set of allowable plans. Obviously, optimal planning algorithms are MIR. In general, non-optimal planning algorithms are not. However, for a number of planning domains approximations are known that can be used to create MIR mechanisms.

In the full version of the paper, we give one such example in the Blocks World (BW) domain. Although *optimal* planning for BW is NP-hard, we propose an MIR algorithm f_{bw} based on the work of [3]. We show that: *(i)* if the set of goals does not contain any conflicts, then (deposit-)VCG-based mechanism using f_{bw} is truthful; *(ii)* if, however, the goals have conflicts, and the social welfare depends on which goals are satisfied, then by limiting the number of goals to be attained by K, we can impose a polynomial bound on the mechanism's time complexity. Thus, a truthful (deposit-)VCG-based mechanism using f_{bw} can be achieved. More generally, *given a polynomial-time algorithm $f_d : \Theta \rightarrow \Pi$ for a planning domain d that is MIR on problems without conflicting goals, and an upper-bound K on the number of goals that is considered, an algorithm f_d^K exists that is MIR and polynomial in the input size*.

For our future work, we are interested in studying how other (approximation) algorithms for planning can be used to construct efficient and truthful mechanisms, focusing especially on variants of existing distributed MAP algorithms.

References

[1] N. Nisan. Introduction to mechanism design (for computer scientists). In *Algorithmic Game Theory*, pages 209–242. Cambridge University Press, 2007.

[2] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *Journal of AI Research*, 29:19–47, 2007.

[3] J. Slaney and S. Thiébaux. Blocks world revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001.

Metrics for Mining Multisets [1]

Walter A. Kosters Jeroen F. J. Laros

Leiden Institute of Advanced Computer Science, Universiteit Leiden, The Netherlands

1 Introduction

A *multiset* (also referred to as a bag) is a set (collection of elements where the order is of no importance), where the elements do not need to be unique. A vase with n blue and m red marbles is a multiset for example.

We propose a new class of distance measures (metrics) designed for multisets, both of which are a recurrent theme in many *data mining* [2] applications. One particular instance of this class originated from the necessity for a clustering of criminal behaviours. Here the multisets are the crimes committed in one year. This metric generalises well-known distance measures like the Jaccard and the Canberra distance.

These distance measures are parameterised by a function f which, given a few simple restrictions, will always produce a valid metric. This flexibility allows these measures to be tailored for many domain-specific applications. The metrics in this class can be efficiently calculated. In the full paper, all proofs are given and various applications are shown.

2 The Metric

In order to produce a decent distance measure d_f, we carefully choose a function $f(x, y)$ that denotes the difference between the number of elements x and y of a specific type. This can not be any function; it has to have a finite supremum M and $f(x, 0)$ must be larger than or equal to $M/2$ (for $x > 0$) in order for the triangle inequality to hold. The function should also be symmetric and $f(x, x)$ should be zero. Also, the triangle inequality must hold for f itself as well. With this f we can now define a metric for multisets. We consider multisets X, Y over $\{1, 2, \ldots, n\}$, and let $x_i \in \mathbb{Z}_{\geq 0}$ (resp. y_i) be the number of times that i ($i = 1, 2, \ldots, n$) occurs in X (resp. Y). For a multiset X, let $S(X)$ denote its underlying set. We define $d_f(\emptyset, \emptyset) = 0$ and for multisets X, Y:

$$d_f(X, Y) = \frac{\sum_{i=1}^{n} f(x_i, y_i)}{|S(X) \cup S(Y)|}$$

if X or Y is non-empty.

The application of weights for certain elements can be done by multiplying the appropriate number of elements by the weight. An important characteristic of these metrics is that the distance increases significantly when we add an extra dimension. This is not the case in other well-known metrics like the standard Euclidean distance.

3 Example

We use the following function to give an impression of the measure. As the expert defined function, we take $f(x, y) = |x - y|/(x + 1)(y + 1)$.

As a test case, we made the following synthetic dataset with fictional crimes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ of increasing severity, and criminals ranging from 1 to 10 as seen in Table 1. For each criminal the number of crimes in

[1] Published in: Research and Development in Intelligent Systems XXIV, Proceedings of AI-2007, the Twenty-seventh SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (M. Bramer, F. Coenen, M. Petridis, editors), Springer, pp. 293–303, Cambridge, UK, 2007.

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

each category is given. For instance, 1 is innocent, 2 is an incidental small criminal, 6 is a one-time offender of a serious crime, and 10 is a severe criminal.

	1	2	3	4	5	6	7	8	9	10
$\mathcal{A}$	0	2	10	0	0	0	0	2	0	2
$\mathcal{B}$	0	0	0	2	0	0	2	4	0	2
$\mathcal{C}$	0	0	0	0	1	0	2	0	3	2
$\mathcal{D}$	0	0	0	0	0	1	1	0	5	2

Table 1: Ten criminals $(1, 2, \ldots, 10)$, four crimes $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$

We use a distance preserving dimension reduction algorithm described in [1] to obtain the visualisations in Figure 1. When we choose the same weights for all categories, we obtain the picture on the left. Number 2 and 3 are close together, and 8 is near there too. This is what we would expect. Number 9, 7 and 5 are close together too, as could be expected; 1, 4 and 6 however are close to each other because they have a large distance to all others.

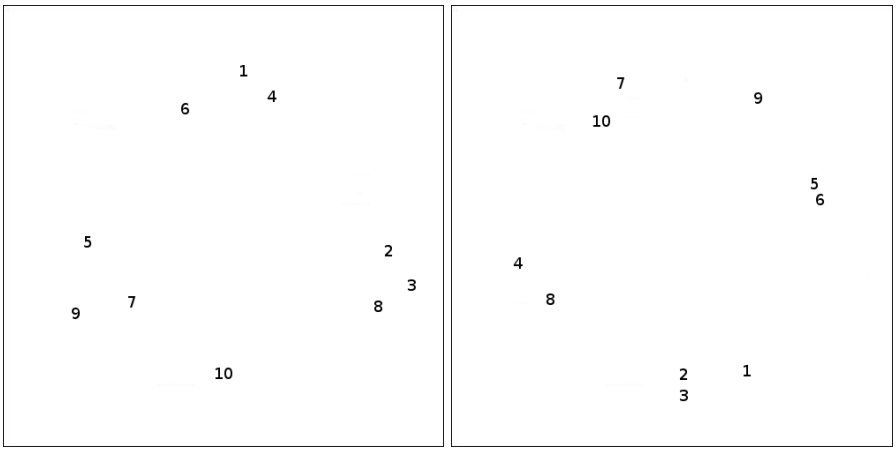


Figure 1: Two different clusterings for ten criminals

If we apply weights $(1, 10, 100, 1000$ for $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ respectively) to accentuate the severity of a crime, we obtain the picture on the right. This gives more insight into the nature of the criminal, 10 and 7 are close together for example (both heavy criminals), while 2, 3 and 1 are close to each other because they have committed no or relatively light crimes.

4 Conclusions and Further Research

We have proposed a flexible distance measure, that is suitable in many fields of interest. It can be fine tuned to a large extent. This may result in different visualisations, illustrating different aspects of the data (see Figure 1).

We can use this measure as a basis for further analysis, like the analysis of criminal careers. In that case, we suggest that the distance measure is used as a basis for *alignment* to make the best match between two careers. By doing this, and by comparing sub-careers, we might be able to extrapolate criminal behaviour based upon the criminal record through time.

References

[1] Kosters, W.A., Wezel, M.C. van, Competitive neural networks for customer choice models, in E-Commerce and Intelligent Methods, volume 105 of Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, 2002, pp. 41–60.

[2] Tan, P.-N., Steinbach, M., Kumar, V., Introduction to data mining, Addison-Wesley, 2005.

A Hybrid Approach to Sign Language Recognition

Jeroen Lichtenauer [a] Emile Hendriks [b] Marcel Reinders [b]

[a] *Imperial College London, 180 Queen's Gate, London SW7 2AZ*
[b] *Delft University of Technology, P.O.Box 5031, 2600 GA Delft*

Abstract

Methods commonly used for speech and sign language recognition often rely on outputs of Hidden Markov Models (HMM) or Dynamic Time Warping (DTW) for classification, which are merely factorized observation likelihoods. Instead, we propose to use Statistical DTW (SDTW) only for warping, while classifying the synchronized features with either of two proposed discriminants. This hybrid approach is shown to outperform HMM and SDTW. However, we have found that combining likelihoods of multiple models in a second classification stage degrades performance of the proposed classifiers, while improving performance with HMM and SDTW. A proof-of-concept experiment, combining DFFM mappings of multiple SDTW models with SDTW likelihoods, shows that also for model-combining, hybrid classification can provide significant improvement over SDTW.

1 Introduction

A recent modification of HMM, called Statistical DTW (SDTW) [1], incorporates the warping flexibility of the exemplar-based DTW in the statistical framework of HMM. Since in [1] SDTW was shown to outperform HMM in on-line handwriting recognition, an improvement over HMM can also be expected when SDTW is applied to sign language recognition. Our results show that this is indeed the case. However, we further improve upon SDTW, based on our main hypothesis:
Proposition *The maximized likelihood that results in the optimal signal warping is not the optimal conditional likelihood estimation of the signal class.*
Instead of relying directly on the likelihoods obtained from (S)DTW or HMM, we consider SDTW primarily as a registration method. The synchronized feature sets are classified by Combined Discriminative Feature Detectors (CDFD) or Quadratic classification of Discriminative Features transformed by a Fisher Mapping (Q-DFFM). Details about these classification methods and the stereo hand tracking can be found in [2].

Our experiments are limited to hand motion trajectories and apparent hand-size change in isolated signs. These are the few components that the current state of the art in human motion analysis allows to track in reasonably soft-constrained situations. We assume that if sign language recognition by motion and hand size change benefits from a hybrid approach, this will certainly be the case if even more parallel aspects (e.g. detailed hand/body pose and facial expression) are considered.

2 Results

Sign classification is evaluated on a set of 120 different signs of the Dutch Sign Language (DSL), each performed by 75 different persons. The images are captured at 640x480 pixels and 25 frames per second. Evaluation is done by 5-fold cross-validation, with a separation of persons between the train and test sets (person-independent), and a separation of non-target classes between train and test sets, in the case of target-class classification (rejection of unseen classes). Our hybrid methods (SDTW + CDFD or Q-DFFM) are compared to SDTW and a 40 state HMM with Bakis topology. Three types of classifications are considered: 1) Target-class classification with a model of the target class, 2) Target-class classification by combining

Table 1: Average classification results for 120 signs and 5 cross-validations. The results for target-class classification are measured in $pAUC_{0.1}$. For multi-class classification, the rate of correct classification is given.

		target-cl. single model	target-cl. multi-m comb.	multi-cl. multi-m comb.
a	HMM	84.61%	96.97%	90.8%
b	SDTW	90.54%	97.22%	90.8%
c	SDTW+CDFD	95.46%	90.86%	76.0%
d	SDTW+Q-DFFM	**96.62%**	94.84%	83.7%
e	SDTW&DFFM5		**97.50%**	**92.3%**

models of multiple classes, 3) Multi-class classification by combining the SDTW models of all classes. In target-class classification, a sign is either detected as the target class, or rejected (binary decision). The rejection threshold determines the trade-off between true positives and false positives (operating point). All possible operating points for the trained classifier of a target class are represented by the Receiver Operating Characteristic (ROC) curve. For evaluation of the trained classifiers, we computed the partial Area Under the Curve (AUC) of the ROC curves, between false positive rates of 0 and 0.1 ($pAUC_{0.1}$).

The first two result columns in table 1 show the average $pAUC_{0.1}$ over the 120 target classes in the 5 cross-validations. Both hybrid methods clearly outperform HMM and SDTW when a single model is used. In the second experiment, the likelihood outputs of multiple (96) models are combined by training a 2nd stage classifier (Fisher) in the output space of all classifiers trained in the previous experiment. Hence, the previous target-class classifier for the real target class, but also target-class classifiers for other classes, are combined together, to classify the real target class. Results are shown in the second results column of table 1. While HMM and SDTW benefit from the model combining, the hybrid methods do not. This is because the target-class classifiers trained with our approach are too specific to be used as inputs for 2nd stage detection of other classes. Because the Fisher Mapping, that is part of Q-DFFM, does contain information about other classes, we combined the SDTW outputs with the first 5 Fisher dimensions of each of the single-model SDTW+Q-DFFM target-class classifiers, increasing the dimensionality for the 2nd stage Fisher classifier by 500% over SDTW alone. This method is indicated by SDTW&DFFM5 in table 1(e). Despite the increase of dimensionality, the richer description increases the partial ROC surface from 97.22% to 97.50%. The significance of this improvement is indicated by a p-value of 0.009 in a paired t-test of $pAUC_{0.1}$ over all individual classifiers. In the multi-class classification experiment, we have combined the single-model target-class classifiers for all 120 sign classes in a single feature space, using Fisher to discriminate between the 120 classes in the 2nd stage. Also here, the SDTW&DFFM5 method outperforms HMM and SDTW significantly (3rd column in table 1). The improvement over HMM has a p-value of 0.019 in a paired t-test of the classification rates over the 5 cross-validation folds. Again, this confirms the benefit of a hybrid approach to sign language recognition.

3 Acknowledgements

This work is a part of the PhD work of Jeroen Lichtenauer, conducted at TU Delft, the Netherlands. Current work of Jeroen Lichtenauer is supported by MAHNOB project of Maja Pantic, funded by the European Research Council under the ERC Starting Grant agreement No. ERC-2007-StG-203143.

References

[1] C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system *frog on hand* and cluster generative statistical dynamic time warping. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 26(3):299–310, March 2004.

[2] J.F. Lichtenauer, E.A. Hendriks, and M.J.T. Reinders. Sign language recognition by combining statistical dtw and independent classification. *Submitted to: Transactions on Pattern Analysis and Machine Intelligence*, 2008.

Improved Situation Awareness for Public Safety Workers while Avoiding Information Overload

Marc de Lignie[a] BeiBei Hu[b] Niek Wijngaards[c]

[a] *vts Politie Nederland, PO Box 238, 3970 AE Driebergen*
[b] *D-CIS Lab / Delft University of Technology, PO Box 90, 2600 AB, Delft*
[c] *D-CIS Lab / Thales Research & Technology, PO Box 90, 2600 AB, Delft*

Extended Abstract

This research stems from the MOSAIC project, a part of the valorization and knowledge transfer effort of the Interactive Collaborative Information Systems (ICIS) research programme (http://www.icis.decis.nl/), supported by the Dutch Ministry of Economic Affairs, grant no.: BSIK03024. ICIS is hosted by the D-CIS Lab (http://www.decis.nl/), the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam and the Netherlands Organization for Applied Scientific Research (TNO).

1 Information Requirements in Crisis Management

Public safety services for emergency aid and disaster management face the challenge of dealing with the enormous growth of information: any information may add to situation awareness and be relevant for handling a specific incident. Evaluations of past incidents and disasters [1] already show too often that the information present in some information system was not brought to the attention of public safety workers in the field or in a (crisis) control room, making them state: "If only I had known… They should have told me!" This challenge provides an interesting case for testing techniques for distributed context-aware information retrieval and dissemination. An emergency incident or disaster provides a structured context in terms of location, time and nature of the incident and roles and tasks of end users.

2 Approach

A new combination of techniques is applied in conjunction, namely (a) Information retrieval techniques such as accessing heterogeneous data, data-fusion and spatiotemporal and semantic link analysis, (b) Filtering techniques for context specific dissemination of information towards mobile workers and (c) Information processing in distributed systems. The multi-agent architecture is shown in Figure 1, which fits in a wider effort to investigate actor-agent communities for decision support in complex and chaotic environments. The following functional layers are distinguished:

- *Data access*: an agent acts as a specific adaptor for a data source. The agent can deal with access and authentication protocols, usage policies, data models, indexing, relevancy ranking, confidentiality, etc.
- *Query distribution and data analysis*: expert agents adapt, split and translate complex queries into more specific queries towards the data access agents and aggregate the returned results. Aggregation may encompass various functions such as ranking, filtering, de-duplicating, correlating with rule bases (threat analysis), etc. During an incident multiple query distribution agents can be invoked, each with its own range of data sources covered and specific expertise to analyze the results.

- *Incident assistance*: Per incident an agent manages the incident information space. This includes proactive gathering and maintaining all information relevant to the incident itself (situation awareness), as well as the influence on other events and vice versa (super situation awareness).
- *Team assistance*: an agent manages the composition of its incident response team as well as filtering and dissemination information to/from individual team members.
- *Personal assistance*: agents are responsible for the human-computer interaction. This includes negotiations with the team assistance agent about the end user attention as well as distributing information and notifications over the available interaction channels (text, audio, vibrations).

This layering realizes a natural separation of concerns and the multi-agent platform supports ad hoc adaptation and integration of techniques and information sources.

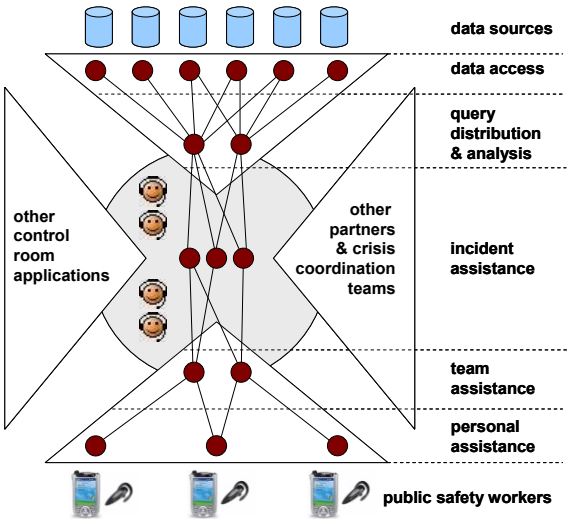


Figure 1. Layers of agents (circles) in the multi-agent system.

The current and ongoing implementation of this multi-agent system is verified by interviews with representatives of Dutch public safety organizations on the basis of progress demonstrations on our research themes:

- Personalized information filtering: what factors from the context and user profiles are important to deliver task-based information to the user while preventing information overload.
- Super situation awareness: this depends on the ability to integrate and process task-relevant information in ways that support decision-making as well as keep track of work context of users.
- Knowledge management: apply domain knowledge to various stages of the information retrieval and dissemination.

3 Conclusions

Dutch public safety organizations acknowledge the need for better use of information available in various sources and reported during the course of an incident. A multi-agent system is suitable for integrating techniques for context-aware information retrieval and dissemination. The ongoing, actual implementation of these techniques for information management in crisis situations requires further research on personalization, knowledge management and the creation of super situation awareness.

References

[1] Advice committee ICT coordination for disaster management (ACIR) 2005, *De vrijblijvendheid voorbij*. Towards effective information provisioning for large scale coordinated actions within our decentralized state, Dutch Ministry of the Interior and Kingdom Relations.

Authorship Attribution and Verification with Many Authors and Limited Data [1]

Kim Luyckx Walter Daelemans

University of Antwerp, CNTS Language Technology Group

Abstract

Most studies in statistical or machine learning based authorship attribution focus on two or a few authors. This leads to an overestimation of the importance of the features extracted from the training data and found to be discriminating for these small sets of authors. Most studies also use sizes of training data that are unrealistic for most situations in which stylometry is applied (e.g., forensics), and thereby overestimate the accuracy of their approach in these situations. In this paper, we show, on the basis of a new corpus with 145 different authors, what the effect is of many authors on feature selection and learning, and show robustness of a memory-based learning approach in doing authorship attribution and verification with many authors and limited training data when compared to eager learners.

1 Introduction

In traditional studies on authorship attribution, the focus is on small sets of authors. Trying to classify an unseen text as being written by one of two or of a few authors is a relatively simple task, which in most cases can be solved with high reliability and accuracies over 95%. The field is however dominated by studies potentially overestimating the importance of the specific predictive features in experiments discriminating between only two or a few authors. A second problem in traditional studies are the unrealistic sizes of training data, which also makes the task considerably easier. Researchers tend to use over 10,000 words per author, which is regarded to be a reliable minimum for an authorial set. When only limited data is available for a specific author (e.g., a letter or e-mail), the authorship attribution task becomes much more difficult. Traditional approaches appeara to be less reliable than expected from reported results when applied to more realistic applications like forensics.

In this paper, we present the first systematic study of the effect of many authors and limited data on feature selection and learning in the tasks of authorship attribution and verification. We show robustness of a memory-based learning approach in doing authorship attribution and verification with many authors and limited training data when compared to eager learners such as SVMs and maximum entropy learning.

2 Corpus and Approach

The 200,000-word Personae corpus in this study consists of 145 student (BA level) essays of about 1400 words about a documentary on Artificial Life, thereby keeping markers of genre, register, topic and age relatively constant. These essays contain a factual description of the documentary and the students opinion about it. The students also took an online Myers- Briggs Type Indicator (MBTI) test and submitted their profile, the text and some user information via a website. The corpus can therefore not only be used for authorship attribution experiments, but also for personality prediction.

We approach authorship attribution as an automatic text categorization task. As in most text categorization systems, we take a two-step approach in which our system (i) achieves automatic selection of features

[1] The full version of this paper appeared in: *Proceedings of the Twenty-Second International Conference on Computational Linguistics (COLING '08), 2008, pp. 513-520.*

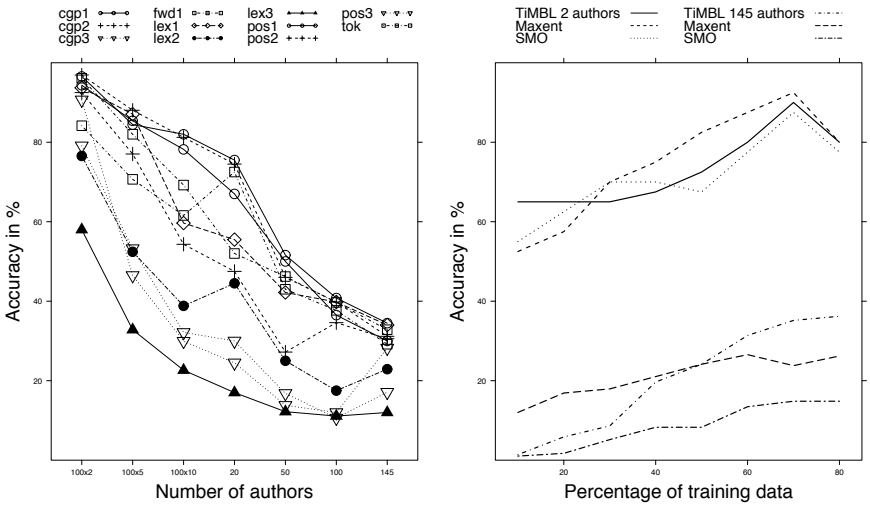
that have high predictive value for the categories to be learned, and (ii) uses machine learning algorithms to learn to categorize new documents by using the features selected in the first stage.

We use the Memory-Based Shallow Parser (MBSP) , for feature construction. MBSP tokenizes the input, performs a part-of-speech analysis, looks for noun phrase, verb phrase and other phrase chunks and detects subject and object of the sentence and a number of other grammatical relations. Features extracted include readability metrics, function word patters, *n*-grams of words and POS tags and vocabulary richness measures. We use the chi-square metric to select constructed features.

3 Experiments and Results (summary)

The paper focuses on three facets of authorship attribution, each with their own experimental set-up: (a) the effect of many authors on feature selection and learning; (b) the effect of limited data in authorship attribution; (c) the results of authorship attribution using many authors and limited data on learning.

For (a), we perform experiments in authorship attribution while gradually increasing the number of authors. We investigate (b) by performing authorship attribution on 2 and 145 authors while gradually increasing the amount of training data, keeping test constant at 20% of the entire corpus. The resulting learning curve will be used to compare performance of eager and lazy learners. The authorship attribution with many authors task (c) - which is closer to a realistic situation in e.g. forensics using limited data and many authors is approached as a one-class learning problem. For each of the 145 authors, we have 80% of the text in training and 20% in test. The negative class contains 80% of each of the other 144 authors training data in training and 20% in test.



(a) The effect of limited data

(b) The effect of many authors

We see, as expected, a marked effect of many authors and limited data in authorship attribution. When systematically increasing the number of authors in authorship attribution, we see that performance drops significantly (Figure a). On the positive side, similar types of features work well for different numbers of authors in our corpus, but generalizations about individual features are not useful. Memory-based learning shows robustness when dealing with limited data, which is essential in e.g. forensics (Figure b). Results from experiments in authorship attribution on 145 authors indicate that in almost 50% of the cases, a text from one of the 145 authors is classified correctly. Using combinations of good working lexical and syntactic features leads to significant improvements. The approximation of the authorship verification task is a much more difficult task, which only leads to a correct classification in 56% of the test cases. It is clear that studies reporting over 95% accuracy on a 2-author study are overestimating their performance and the importance of the features selected.

Further research with the 145-author corpus will involve a study of handling with imbalanced data and experimenting with other machine learning algorithms for authorship attribution and verification and a more systematic study of the behavior of different types of learning methods (including feature selection and other optimization issues) on this problem.

Agent Performance in Vehicle Routing when the Only Thing Certain is Uncertainty

Tamás Máhr [ab] Jordan Srour [c] Mathijs de Weerdt [a] Rob Zuidwijk [c]

[a] *Delft University of Technology, Delft, The Netherlands*
[b] *Almende BV, Rotterdam, The Netherlands*
[c] *RSM Erasmus University, Rotterdam, The Netherlands*

1 Introduction

Scheduling the routes of trucks to pick-up and deliver containers is a complex problem. In general such Vehicle Routing Problems (VRPs) are known to be NP-complete, and therefore inherently hard and time consuming to solve to optimality. Fortunately, these problems have a structure that can facilitate efficient derivation of feasible (if not optimal) solutions. Specifically, the routes of different trucks are more or less independent. Such "locality" in a problem is a first sign that an agent-based approach may be viable. We think it is safe to assume, based on its long history, that current global-optimization practice in operations research (OR) outperforms local agent-based approaches in settings where all information is known in advance (static settings). However, in situations with high uncertainty, agent-based approaches are expected to outperform these traditional methods [3].

In this paper we investigate whether a distributed agent-based planning approach indeed suffers less from job arrival uncertainty than a centralized optimization-based approach. In order to compare the two different approaches, we use the best available algorithms for both sides.

2 Agent-based Approach

Since we are primarily interested in distributed agent models, we use an uncompromisingly flat architecture: no agents can concentrate information from a multitude of other agents. Our agents use a combination of existing techniques to compute the solution. The basic model is similar to that of Fischer et al. [1]. Order agents hold auctions in order of their arrival, and truck agents bid in these auctions. Every truck agent submits a bid that reflects its cost associated with transporting the given order. This cost is a quantity in the time domain. To calculate it, a truck considers *inserting* the new order into its plan, or alternatively *substituting* one of the already contracted orders by the new one. Substitution draws motivation from leveled commitment contracts as described by Sandholm and Lesser [5], and applied by 't Hoen and La Poutré in a transportation problem [2]. To clear an auction, order agents choose the best bid as winner, and respond positively to the winner and negatively to the others. For this we chose a one-shot auction (and more specifically, a Vickrey auction) for its computational efficiency.

In addition to bidding on auctions for new orders, truck agents have another way to enhance the overall solution. At random time intervals, every truck randomly selects an order in its plan and releases it. An order agent that is released (just as those order agents that are substituted) initiates a new auction to find another place. In most cases, these auctions result in the very same allocation as before the release. Nevertheless, sometimes they do manage to find a better place and make a contract with another truck. This implements a distributed improvement technique, similarly to Kohout and Erol, who implemented a distributed order-swapping protocol in their logistical multi-agent system [4].

3 On-line Optimization Approach

To estimate the value of the agent-based solution approach, we study it in comparison to two optimization based solution approaches: (i) a mixed-integer program for solving the static *a priori* case in order to provide a baseline benchmark, and (ii) an on-line optimization approach, which can solve the same dynamic problem instances are the agent approach, and designed to represent current vehicle routing decision support systems.

At the core of both the static *a priori* solution and the on-line optimization is a mixed integer program (MIP) for a truck-load vehicle routing problem with time windows [6], which is given to CPLEX. In the static *a priori* approach, we solve the MIP using all available information, including the dynamic order releases. This is similar to computing at the end of the day what would have been the best to do. In our on-line approach, the MIP is invoked periodically at fixed intervals. At each interval, the full and current state of the world is encoded in the MIP, and solved via a call to CPLEX. The solution given by CPLEX is parsed and any jobs that are within two intervals of being late are permanently assigned. Any jobs that were designated for rejection in the solution are rejected only if they are within two intervals of violating a time window; otherwise they are considered available for scheduling in a subsequent interval. The procedure continues in this fashion until the end of the working day at which point all jobs have been served or rejected.

4 Results

The instances we used for the comparison was based on data provided to us by a Dutch logistics service provider. We rendered their data into 26 days with four separate scenarios of varying levels of order arrival uncertainty. In every scenario, different amount of orders were released during the day, while the rest at the beginning of the day. In one of the extreme scenarios every order was released at the beginning of the day, and in the other extreme scenario, all orders were released during the day.

Running the *a priori*, the on-line optimization, and the agent-based methods on the four instances and averaging over 26 days revealed that the on-line optimization method outperforms the agent-based method when all orders were known at the beginning of the day. Contrarily, when fifty percent or more of the orders were dynamic (released during the day), agent-based methods performed competitively.

This comparison study of agent-based and centralized vehicle routing approaches featured dynamic order releases, and a uniform distribution of time windows. It is left for future work to compare the methods on time-window distributions with bursts, and alternatively use other sources of uncertainty, such as travel-time or loading/unloading-time uncertainty.

References

[1] Klaus Fischer, Jörg P. Muller, Markus Pischel, and Darius Schier. A model for cooperative transportation scheduling. In *Proc. of the 1st Int. Conf. on Multiagent Systems*, pages 109–116, Menlo park, California, 1995. AAAI Press / MIT Press.

[2] Pieter Jan't Hoen and Johannes A. La Poutré. A decommitment strategy in a competitive multi-agent transportation setting. In *Proc. of 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 1010–1011, New York, NY, USA, 2003. ACM Press.

[3] NR Jennings and S. Bussmann. Agent-based control systems: Why are they suited to engineering complex systems? *Control Systems Magazine, IEEE*, 23(3):61–73, 2003.

[4] Robert Kohout and Kutluhan Erol. In-time agent-based vehicle routing with a stochastic improvement heuristic. In *AAAI '99/IAAI '99: Proc. of the 16th national conference on Artificial Intelligence and the 11th on Innovative Applications of Artificial Intelligence*, pages 864–869, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[5] Tuomas W. Sandholm and Victor R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behaviour*, 35:212–270, 2001.

[6] J. Yang, P. Jaillet, and H. Mahmassani. On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record*, 1667:107–113, 1999.

Design and Validation of HABTA: Human Attention-Based Task Allocator (Extended Abstract)[1]

Peter-Paul van Maanen [ab] Lisette de Koning [a] Kees van Dongen [a]

[a] *TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands*
[b] *Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands*

Several challenges can be identified for work on future naval platforms. Information volumes for navigation, system monitoring, and tactical tasks will increase as the complexity of the internal and external environment also increases. The trend of reduced manning is expected to continue as a result of economic pressures and humans will be responsible for more tasks, tasks with increased load, and tasks with which they will have little experience. Problems with attention allocation are more likely to occur when more has to be done with less. To avoid these attention allocation problems, in this paper it is proposed that humans are supported by cooperative agents capable of managing their own and the human's allocation of attention. It is expected that these attention managers have a significant impact: when attentional switches between tasks or objects are often solicited, where the human's lack of experience with the environment makes it harder for them to select the appropriate attentional focus, or where an inappropriate selection of attentional focus may cause serious damage. In domains like air traffic control or naval tactical picture compilation these properties are found.

The main contribution of the present paper is the description of the combined approach of design and validation for the development of applied cooperative agent-components. The design requirements are given of an agent-component called *Human Attention-Based Task Allocator (HABTA)*. This component enables the agent to support the human-agent team by managing attention allocation of the human and the agent. The HABTA-component does this by reallocation of the human's and agent's focus of attention to tasks or objects based on an estimation of the current human allocation of attention and by comparison of this estimation with normative rules. In Figure 1 the design overview of a HABTA-component is shown that corresponds to the above mentioned design requirements. The setting in this particular overview is a naval officer behind an advanced future integrated command and control workstation and compiling a tactical picture of the situation. If the agent cooperatively assists the officer, than the agent should have a descriptive (Requirement 1) and normative model (Requirement 2). When the operator allocates his attention to certain objects or tasks that also require to receive attention, the outcome of both models should be comparable. This means that output of the models should not differ more than a certain threshold. The output of the two models in the example shown in Figure 1 are clearly different: in the top-left image, the operator is attending to different objects and corresponding tasks than the top-right image indicates as being required (see red arrows). Because of this discrepancy, which the HABTA-component should be able to determine (Requirement 3), an adaptive reaction by the agent is triggered (Requirement 4). This means that, for instance, the agent either will draw attention to the proper region or task through the workstation, or it will allocate its own attention to this region and starts executing the tasks related to that region, for the given situation.

Requirements 1–4 are constrain the HABTA-component architecture, but in order to make it effective, two types of questions should be answered: 1) are the descriptive and prescriptive models accurate enough, and 2) does the use of the HABTA-component based on those models lead to significant improvements? These two questions can be translated into two complementary validation experiments. The first experiment

[1] This paper is an extended abstract of a paper published in Constructing Ambient Intelligence: AmI-07 Workshops Proceedings, Communications in Computer and Information Science (CCIS) [2].

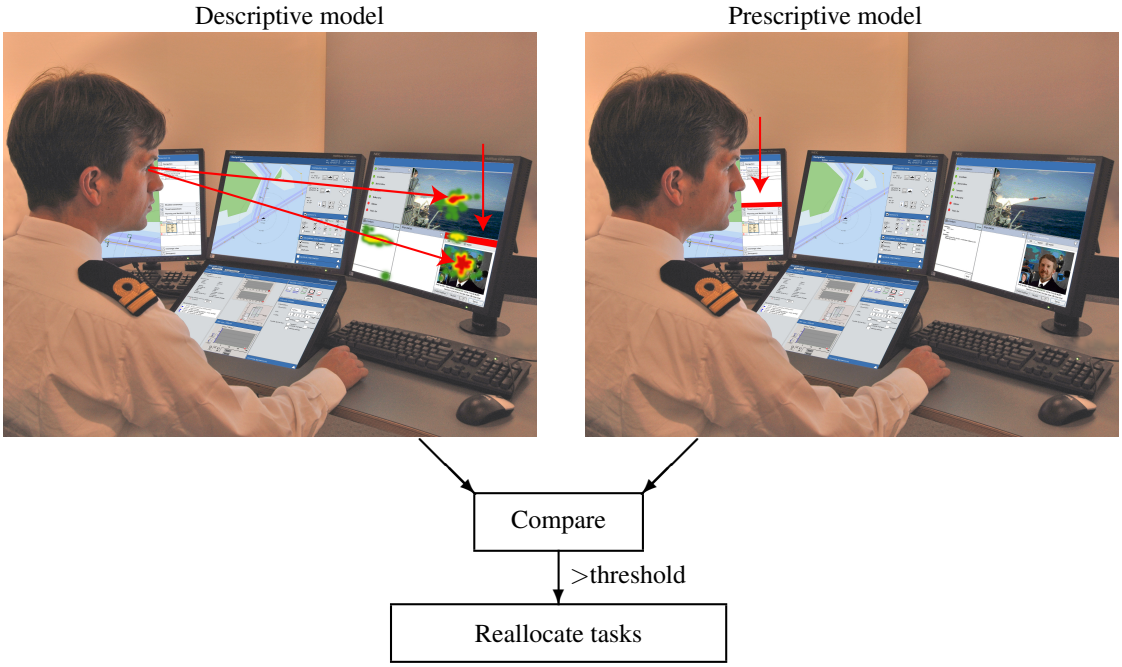


Figure 1: Design overview of HABTA-component for future integrated command and control environment.

validates the descriptive model, where validity is calculated by comparison of the model's and human's estimate of human attention allocation. The amount of overlap signifies the validity of the model. The second experiment validates the HABTA-component itself, measuring its effect in terms of human-agent team performance, trust, and reliance. Intermediary results suggest that the HABTA-component can be useful, but that more experiments have to be performed. Therefore, future research will focus on the performance and analysis of these experiments. It is expected that the accuracy of the model can be increased hereafter, however 100% accurateness will not be attainable. The results of the first experiment will show if the variables indeed provide enough information to improve the accurateness of the model.

If the agent does not support the human at the right time and in the right way, this might influence trust and acceptance of the agent. It is interesting to investigate whether an observable and adjustable internal structure of the agent improves trust and acceptance of the system (e.g. [1]). This also needs further research.

In this paper the development and validation of a normative model (prescriptive model) is not described. Validation of this model is important, as it is also a crucial part of the HABTA-component. Errors in this model will lead to support at the wrong time and this will influence performance, trust, and acceptance. Further research is needed in order to develop and validate normative models.

Finally, though a solution for the problem of human error in the allocation of attention in the chosen domain is important, in general, agent-components have more value when they can be easily adjusted for other applications too. It is therefore interesting to see whether HABTA-based support can be applied in other domains as well. The results of the planned experiments are expected to give clues about the scalability of the presented approach.

References

[1] Tina Mioch, Maaike Harbers, Willem A. van Doesburg, and Karel van den Bosch. Enhancing human understanding through intelligent explanations. In T. Bosse, C. Castelfranchi, M. Neerincx, F. Sadri, and J. Treur, editors, *Proceedings of the first international workshop on human aspects in ambient intelligence*, 2007.

[2] Peter-Paul van Maanen, Lisette Koning de, and Kees van Dongen. Design and validation of HABTA: Human attention-based task allocator. In M. Mühlhäuser, A. Ferscha, and E. Aitenbichler, editors, *Constructing Ambient Intelligence: AmI-07 Workshops Proceedings, Communications in Computer and Information Science (CCIS)*, volume 11, pages 286–300. Springer Verlag, 2008.

Improving People Search Using Query Expansion: How Friends Help to Find People

Thomas Mensink[a] Jakob Verbeek[b]

[a]*SNN, Radboud Universiteit, Nijmegen*, [b]*LEAR team, INRIA Rhône-Alpes, Grenoble*
T.Mensink@science.ru.nl, Jakob.Verbeek@inria.fr

Abstract

In this paper we are interested in finding faces of specific people in a large database of captioned news images. It has recently been shown that analysis of the faces in images returned on a text-based query over captions can significantly improve search results. The contribution of this paper is to improve these search results by exploiting faces of other people that co-occur frequently with the queried person. In the face analysis we use the query expansion to provide a query-specific relevant set of 'negative' examples which should be separated from the potentially positive examples in the initial result set. We experimentally evaluate our proposed methods using a set of 23 queries on a database of 15.000 captioned news stories from *Yahoo! News*. The results show that query expansion improves the performance, and current state-of-the-art results are outperformed by 10% precision on average. This is an abstract of our paper [1].

1 Introduction

Over the last decade we have witnessed an explosive growth of image and video data available both on-line and off-line. In this paper we aim to improve search for people in databases of captioned news photographs. Using only the information of the captions (i.e. simply returning all images with the name of the queried person in the caption) leads to a disappointing precision of 44% (fraction of faces belonging to the queried person among all returned faces, averaged over queries for 23 people).

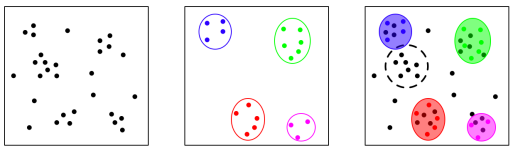
Recently several authors have shown that initial text-based results can be significantly improved by filtering faces on the basis of visual features. The idea underlying this clean-up of text-based results is that the queried person will appear relatively often compared to other people, so we can search for a large group of highly similar faces among those returned by a text-based search. The performance of these methods depends strongly on this assumption: for people whose face represents less than about 40% of faces in the initial result set, performance may be very poor.

Resolving all face-name associations in a database of captioned news images has also been considered. The potential advantage of solving name-face associations for multiple names at once is that the faces associated with one name may resolve ambiguities for other names. However, a drawback is the combinatorial number of possible assignments between faces and names. Therefore optimization of parameters to the best association is a tedious task and prone to get stuck at local optima.

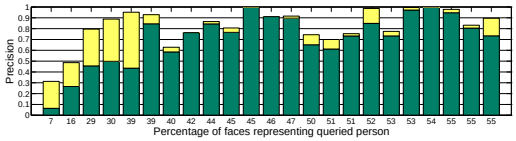
2 Query Expansion

We explore the middle ground between solving the complete name-face association problem, and analysis of only the initial result set. A query is expanded to names that co-occur often with the queried person. With this 'query-expansion' we obtain a notion of whom we are *not* looking for. We find that query expansion gives dramatic improvements in the failure mode of existing work.

By querying the database for names that frequently appear together with the queried name we can collect a set of faces that help understand whom we are *not* looking for. For example, suppose that in captions for the query *Tony Blair* the names *George Bush* and *Gordon Brown* occur often. We can then query the database



(a) The distribution of face features found using the initial text-based query (left), the query expansion composed of additional faces found using queries for four different people that co-occur (middle), and how the query expansion helps to identify faces in initial result set that are not the queried person (right).



(b) Precision obtained with the generative model, when using the query to fit the background model (green/dark bars), and precision increase when using query expansion (yellow/bright bars). The 23 queries are sorted by the text-based precision for each query which is indicated on the horizontal axis.

Figure 1: (a) Illustration of query expansion, and (b) performance increase using query expansion.

for images with *George Bush* and *Gordon Brown* in the caption, but without *Tony Blair*. Using this query expansion, we can then rule out faces in the initial text-based result set that are very similar to many faces returned for *George Bush* or *Gordon Brown*. See Figure 1(a) for a schematic illustration of the idea.

We apply the idea of query expansion to a generative mixture model to filter the text-based results, and also to a linear logistic discriminant method to filter the initial results. The discriminative method is based on a binary classifier that classifies faces of one person against other faces. The generative method associates one Gaussian in the feature space with each person (the queried person and co-occurring people), and models faces of all other people using a single broad Gaussian.

3 Experimental Results and Conclusions

We evaluated our proposed methods on a set of 23 queries also used in previous work. Searches are performed on a data set of about 15.000 captioned news photographs downloaded from *Yahoo! News*. For each query a ground-truth labeling was made manually.

The images of the database are processed using a face detector and a facial feature detector (returning 13 locations relative to the eyes, nose, and mouth), and lighting is normalized with a Difference of Gaussian method. For each of the facial feature points a SIFT descriptor is calculated, yielding a feature vector for each face. Captions are scanned with a named entity detector to find frequently co-occurring names for the query expansion. The text-based queries are solved by simple string matching against all captions.

In Figure 1(b) we show the performance of the Gaussian mixture approach without query expansion and with query expansion for the 23 queries. Query expansion leads to significant performance increases, especially in cases where the percentage of faces in the initial text-based result set that represent the queried person is low.

Query expansion is particularly useful for the Gaussian mixture approach to filter the initial query result. Without query expansion the discriminative method clearly outperformed the generative one ($\approx 10\%$ difference in precision), but these differences become smaller using query expansion ($\approx 5\%$ differences in precision). From a practical point of view the generative model with query expansion may be preferred because of its ease of implementation, and because it is much faster to train than the discriminative model.

We have achieved performance levels that are significantly higher than those obtained in previous work using similarity-based methods. Our best method (discriminative + expansion) obtains a precision of 87% (84%) for a recall of 75% (85%), while the best previously reported result on these queries only reaches 78% (73%) for the same recall values.

References

[1] Thomas Mensink and Jakob Verbeek. Improving people search using query expansions: How friends help to find people. In *10th European Conference on Computer Vision*, 2008.

The tOWL Temporal Web Ontology Language

Viorel Milea Flavius Frasincar Uzay Kaymak

Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062 PA Rotterdam

The full version of this paper, entitled KNOWLEDGE ENGINEERING IN A TEMPORAL SEMANTIC WEB CONTEXT appeared in: Proceedings of the Eighth International Conference on Web Engineering (ICWE 2008), pages 65-74, IEEE Computer Society Press, 2008.

Abstract

The emergence of Web 2.0 and the Semantic Web as established technologies is fostering a whole new breed of intelligent Web applications and systems. These are often centered on knowledge engineering and context awareness. However, adequate temporal formalisms underlying context awareness are currently scarce. Our focus in this paper is two-fold. We first introduce a new OWL-based temporal formalism - tOWL - for the representation of time, change, and state transitions. Based hereon we present a financial Web-based application centered on the aggregation of stock recommendations and financial data.

1 Introduction

Information on the Web is mostly textual in nature. Its character is descriptive and meaningless without its most skilled interpreter: the human brain. If the goal of automating the aggregation of vast amounts of information is to be achieved, then this information should be described in a machine-readable way enabling applications to at least simulate some understanding of the data being processed. The emergence of Web 2.0 and the Semantic Web as established technologies is fuelling a transition from a Web of data to a Web of knowledge. In turn, this knowledge rich environment is fostering a whole new breed of intelligent Web applications and systems, centered on knowledge aggregation and context awareness. Focusing on the latter, it can rightfully be stated that enabling context awareness involves the existence of adequate temporal formalisms - currently very scarce in a Semantic Web context. This results in ad hoc (and often not reusable) solutions for dealing with temporal aspects on the Web.

One of the domains with a prominent temporal aspect, which forms the focus of our current research, is the financial one. More specifically, we seek to explore the area of engineering Web applications for automated trading, an area far too little investigated in such a context. Although seemingly not directly related to automated trading, the Semantic Web may come to meet the increased technological demands emerging in the world of trading. In achieving this purpose, it is necessary to provide extensions to current Semantic Web languages, thus making the latter more suitable for the knowledge we seek to represent. One such extension is presented in this paper and concerns a temporal ontology language based on OWL-DL - the TOWL language. This language stands at the basis of the financial application we present in this paper, and forms one of the key ingredients allowing the aggregation of historical stock recommendations and financial data.

2 The Approach

The temporal language we propose is designed as a set of extensions built on top of OWL-DL. The complete tOWL layer cake is presented in Figure 1. As can be observed from this figure, the foundation for our current approach consists of the DL species of OWL, upon which time-related functionality is added. The first extension introduced by tOWL concerns the expressiveness of the language in a broader sense, rather than being restricted to a temporal domain. Based on current results on concrete domains [1], tOWL has a *Concrete Domains Layer* (CDL) that enables the representation of restrictions over pairs of

compositions of functional roles (feature chains) as well as restrictions over pairs of compositions of a functional and a non-functional role.

Partly enabled by the CDL, the *Time Representation Layer* (TRL) adds a temporal reference system to the language, in the form of concrete time (instants and intervals) and 13 concrete temporal relations – Allen relations (e.g., 'equal', 'before', 'after', 'meets', 'met-by', etc.). The TRL employs the CDC layer for the representation of temporal restrictions between time-bounded entities, as for example the interval concept is defined as having the start time instant before or equal to the end time instant.

Upon enabling temporal reference in the language, the representation of change and state transitions is provided through the *TimeSlices/Fluents Layer* (TFL). This extension enables the representation of temporal parts of individuals, through timeslices and fluents [2]. Timeslices represent the temporal parts of concept instances. Fluents are properties associated to concepts that may have differerent values at various moments in time.

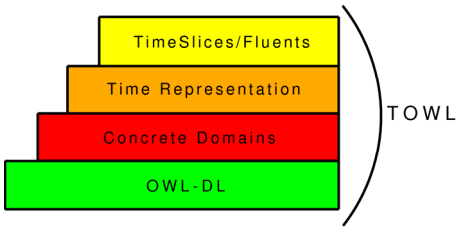


Figure 1. The tOWL Layer Cake

3 Stock Recommendations Aggregation System

Stock recommendations, although taking on different denominations, can always be reduced to advices of the form buy/hold/sell. They are issued by large brokerage firms, and mirror the expectations regarding the development of the stock price of the envisioned company. The collection of such recommendations that are true at a given point in time is denoted as market consensus, and can often be a good indicator of the average expectation regarding the future (within 1 year) value of a company. Roughly, a stock recommendation thus consists of the issuer (the brokerage firm), the targeted company, and the type of the advice (buy/hold/sell).

As a proof-of-concept we have implemented a Stock Recommendations Aggregation System (SRAS), which uses tOWL to represent and reason with the temporal dimension of emerging stock recommentions. The preliminary results present a number of interesting aspects. Perhaps the most striking one relates to the fact that the aggregated recommendation generated by SRAS does not always follow the market consensus. In other words, a recommendation's distribution (across buy, hold and sell) that has a unique maximum (say buy), does not always agree with the advice generated by the application. This could be an indicator that taking into account historical performance of brokerage firms leads to the creation of new knowledge regarding the most likely development of a company's share price.

4 Acknowledgement

The authors are partially supported by the EU funded IST STREP Project FP6 – 26896.

References

[1] C. Lutz and M. Milicic. A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes. *Journal of Automated Reasoning*, 38(1-3):227-259, 2007.

[2] C. Welty and R. Fikes. A Reusable Ontology for Fluents in OWL. In *Proceedings of the Fourth International Conference (FOIS 2006)*. Frontiers in Artificial Intelligence and Applications, vol. 150, pages 226-236, IOS Press, 2006.

A Priced Options Mechanism to Solve the Exposure Problem in Sequential Auctions[1]

Lonneke Mous Valentin Robu Han La Poutré

CWI, Dutch National Research Center for Mathematics and Computer Science
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands

1 Introduction

Auctions are considered to be an important part of the growing electronic commerce, as they can efficiently allocate goods and resources between agents. An important practical issue in the deployment of auction mechanisms is that bidding agents may desire bundles of items, which in many real-life market settings are sold sequentially or in simultaneously ascending auctions. Whenever a buyer agent can obtain synergy between these items, he faces the exposure problem. Informally, the exposure problem occurs whenever an agent may buy a single item at a price higher than what it is worth to him, in the hope of obtaining extra value through synergy with another item, sold later. However, if he then fails to buy the second item at a profitable price, he incurs a loss.

The main way to address this problem in existing e-commerce literature is to replace sequential allocation with a one-shot, combinatorial auction. Combinatorial auction mechanisms proved successful in theory and practice, but they do have some important limitations. The first of these limitations is computational: they typically require a central point of authority to receive all the bids and compute the optimal allocation. Furthermore, many allocation problems occurring in practice are inherently decentralized and sequential. Different sellers may prefer, for a variety of reasons, to sell their items separately - or even through different markets, as the number of electronic auction sites online indicates. Finally, in many application settings (such as distributed logistics), the resources to be allocated are not all known in advance, but they appear dynamically over time. In our work, we take a different approach, which preserves the sequential, decentralized structure of the market, but involves auctioning *options* for the items, instead of the items themselves.

2 Options as a solution: an overview

An option is a contract between the buyer and the seller of a item, where the buyer has the right to choose in the future whether or not he will purchase the item against the pre-agreed *exercise price*. The seller is then bound to sell the item at the demand of the buyer. Since the buyer gains a right, he has to pay the *option price* regardless of whether he will exercise the option or not.

Options help a synergy buyer with the exposure problem. He still has to pay the option price, but if he fails to complete his desired bundle, then he does not pay the exercise price as well, and thus he limits his loss. The risk of not winning subsequent auctions is partly transferred to the seller, who may miss out on the exercise price. However, the seller can benefit indirectly from additional synergy buyers participating in the market, who would have otherwise stayed out. Priced options have a long history of study in finance (see [1] for an overview). However, the option pricing models employed in financial markets usually depend on an underlying asset and assume its true value moves independently of the local actions of individual agents (e.g. this motion is assumed to be Brownian for Black-Scholes models). This crucial assumption does not hold for the online, sequential auctions considered in most e-commerce literature.

[1] This paper is an extended abstract of work presented in [4, 3].

The first work to introduce an explicit option-based mechanism for sequential-auction allocation of items to the MAS community is [2]. They create a market design with free (i.e., zero-priced) options, in which buyers place their bid through proxy agents provided by the mechanism. They show that, in this market design, truth-telling is a dominant strategy on the part of the buyers. The sellers are incentivised to use the proposed options mechanism by market entry effects. However, there may be cases when the market entry effects are not sufficient to motivate the sellers to offer options for free (due to the risk of remaining with their items unsold). In such cases, only positively-priced options can provide sufficient incentive for both sides to use the mechanism. Furthermore, the design proposed in [2] may fail when several synergy buyers are active in the market simultaneously.

By comparison, in our recent work [4, 3] we take a slightly different approach, in which the sellers do cover a part of their risk through an option price. Since an option as defined by two prices (option price + exercise price), an adjustment needs to be made to the standard auction with bids of a single price. We consider a model in which buyers obtain the right to buy the item at a future time, for a pre-determined exercise price. The seller determines the exercise price of an option for the item he has for sale and then sells this option through a first-price auction. Buyers bid for the right to buy this option, i.e., they bid on the option price. Note that in this model, direct auctions appear as the particular case of fixing the exercise price at zero: such options would always be exercised, assuming free disposal.

Our approach and analysis can be characterized as decision-theoretic, meaning both buyer and seller reason with respect to expected future prices. In [4], we take a two-step approach. First, we consider a setting in which n complementary-valued items are auctioned sequentially, assuming there is only one synergy buyer (the competition consists of local bidders desiring only one item). In such a direct, sequential auction, the buyer may bid less than his true value, due to the uncertainty of acquiring future items in his desired bundle. If, instead, options are sold for these items, the agent may bid an option price corresponding to a higher total amount (option + exercise price) than in a direct sale, because he does not have to pay the exercise prices if he fails to get the desired combination. However, the seller takes an exposure risk by auctioning options instead of items (that of not collecting the exercise price), and in order for him to have an incentive to offer options, he expects an increase in the bids he receives, greater than some minimum bound. For this restricted setting, we were able to show analytically that using priced options can increase the expected profit for both the synergy buyer and the seller, compared to auctioning the items directly. We were also able to derive analytically the equations for the minimum and maximum bounds between which the bids of the synergy buyer should fall, in order for both sides to have an incentive to use options. In the second part of the paper, we experimentally study market settings in which multiple synergy buyers are active simultaneously. In such settings, the problem of fixing the right exercise price becomes harder, because the seller has to maximize expected buyer participation, but at the same time reduce his own exposure. While some synergy buyers lose because of the additional competition, others may actually benefit, because sellers have an incentive to fix exercise prices at levels which encourage participation of more buyers.

To conclude, we showed that, while priced options do not always completely eliminate the exposure problem, they can significantly reduce it. The use of options instead of direct sale can increase the expected payoff from participating in a sequential-auction market for both buyers and sellers. This encourages market participation of buyers with complementarities who would otherwise stay out and, thus, increases the allocation efficiency of the market.

References

[1] John C. Hull. *Options, Futures, and Other Derivatives, 5th ed.* Prentice Hall, 2003.

[2] Adam I. Juda and David C. Parkes. An options-based method to solve the composability problem in sequential auctions. In *Proc. of the 7th International Workshop on Agent-Mediated Electronic Commerce*, pages 44–58. Springer LNAI, 2006.

[3] Lonneke Mous, Valentin Robu, and Han La Poutré. Can priced options solve the exposure problem in sequential auctions? In *ACM SIGEcom Exchanges, vol. 7(2)*. ACM Press, 2008.

[4] Lonneke Mous, Valentin Robu, and Han La Poutré. Using priced options to solve the exposure problem in sequential auctions. In *Proc. of 10th International Workshop on Agent-Mediated Electronic Commerce (AMEC'08), Estoril, Portugal*. Springer LNAI (to appear), 2008.

Autonomous Scheduling with Unbounded and Bounded Agents[1]

Chetan Yadati Narasimha[a] Cees Witteveen [a] Yingqian Zhang [a]

Mengxiao Wu [b] Han la Poutré [b]

[a] *Delft University of Technology, Delft*
[b] *Centrum voor Wiskunde en Informatica, Amsterdam*

Autonomous scheduling deals with the problem of enabling agents to schedule a set of interdependent tasks in such a way that their individual schedules always can be merged into a global feasible schedule without suffering any changes. Unlike the traditional approaches to distributed scheduling we do not enforce a fixed schedule to every participating agent. Instead we guarantee *flexibility* by offering a set of schedules to choose from in such a way that every agent can choose its own schedule independently from the others. Furthermore, we study the *performance loss* on efficiency due to such flexibility of the agents.

We assume a finite set of tasks T, each of which takes finite time $d(t_i) \in Z^+$ to complete. Furthermore, these tasks are interrelated by a partially ordered precedence relation $\prec$, where $t_i \prec t_j$ indicates that t_i must be completed before t_j can start. We use the transitive reduction $\ll$ of $\prec$ to indicate the immediate precedence relation between tasks, i.e., $t \ll t'$ iff $t \prec t'$ and there exists no t'' such that $t \prec t''$ and $t'' \prec t'$. We use a directed acyclic graph (DAG) $G = (T, \ll)$ to represent the task structure of T. We assume that T has been assigned to a set of autonomous agents A according to a pre-defined task allocation $\phi : T \rightarrow A$. We denote the set of tasks allocated to agent A_i by $T_i = \phi^{-1}(A_i)$. We also assume that there is a function $c : \{1, 2, \ldots n\} \rightarrow Z^+ \cup \{\infty\}$ assigning to each agent A_i its concurrency bound $c(i)$. This concurrency bound is the upper bound on the number of tasks agent A_i is capable of performing simultaneously. We say that $\langle \{T_i\}_{i=1}^n, \prec, c(), d() \rangle$ is a *scheduling instance*. Given such a scheduling instance, a global schedule for it is a function $\sigma : T \rightarrow Z^+$ determining the starting time $\sigma(t)$ for each task $t \in T$. Of course, we prefer a schedule σ which minimizes *makespan*, i.e., among all feasible schedules σ', we prefer a schedule σ such that $max_{t \in T}\{\sigma(t) + d(t)\} \leq max_{t \in T}\{\sigma'(t) + d(t)\}$.

Our goal is to design a minimal set of constraints C_i for each agent A_i, such that the merging of individual schedules σ_i that satisfy C_i always is a globally feasible schedule. Moreover, we would like the merging to be also *makespan efficient*.

Agents with unbounded concurrent

We show that there exists a simple makespan efficient autonomous scheduling, provided that the agents are capable to process as much tasks concurrently as possible, i.e., they have unbounded concurrent capacity.

The algorithm is based on determining the earliest and latest starting times for a task in a given problem instance. To determine the earliest and the latest starting times for a task we first determine the *depth* and *height* of the task in the task graph. Based on these quantities we determine an interval $[lb(t), ub(t)]$, where $lb(t) = depth(t)$ and $ub(t) = height(t)$. These quantities alone are not enough to ensure autonomous scheduling because, tasks constrained by precedence relationships might have overlapping intervals. In such cases, it is possible for agents to violate precedence constraints between the tasks and hence, we arbitrate such overlaps between any two tasks t, t' with the precedence relation $t \prec t'$ as follows: $ub(t) = lb(t) + \lfloor \frac{ub(t') - lb(t) - d(t)}{2} \rfloor$ and $lb(t') = ub(t) + d(t)$

The complete description of the algorithm is given in the Algorithm 1.

[1]The full version of the paper will appear in the Proceedings of Sixth German Conference on Multi-Agent system Technologies (MATES'08).

Algorithm 1 Generalised Interval based Schedulinge (ISA)

Require: Partially ordered set of tasks $(T, \prec)$, for every task $t \in T$ its depth $depth(t)$ and its height $height(t)$;
Ensure: For every $t \in T$ its scheduling interval $C(t) = [lb(t), ub(t)]$;
1: $depth(T) := max_{t \in T}\{depth(t) + d(t)\}$
2: **for all** $t \in T$ **do**
3: $lb(t) := depth(t)$
4: $ub(t) := depth(T) - height(t)$
5: **end for**
6: **for all** $t, t' \in T$ such that $t \prec t'$ and $lb(t') - ub(t) < d(t)$ **do**
7: $ub(t) = lb(t) + \lfloor \frac{ub(t') - lb(t) - d(t)}{2} \rfloor$
8: $lb(t') = ub(t) + d(t)$
9: **end for**
10: **for all** $t \in T$ **do**
11: return $C(t) = [lb(t), ub(t)]$
12: **end for**

We show that: *(i)* ISA algorithm ensures a *correct* global schedule and it is *efficient* in terms of makespan, and furthermore, *(ii)* ISA gives *maximum* flexibility to agents without sacrificing makespan efficiency.

Agents with bounded concurrent

Algorithm ISA works with the assumption that agents have a capacity to perform an unbounded number of tasks at any given point of time. This assumption is in several cases quite unrealistic. Hence, we adapt the method to accommodate for such bounded concurrency requirements of agents. In particular, we consider the case where agents are strictly *sequential*.

However, it can be shown that if we limit the capacity of agents to performing a single task at any point in time, then given a sequential scheduling instance and a positive integer M, the problem to decide whether there exists a set of constraints C such that the scheduling instance allows for a solution with makespan M by autonomous scheduling is NP-hard. This can be proved through reduction to the well known Set Partitioning problem. Therefore, we have to rely on approximation algorithms for autonomous scheduling in the sequential agent case. Our proposed algorithm first uses the ISA algorithm to determine the set of constraints C for the unit duration tasks. As we have shown above, if the agents would be able to handle tasks concurrently, an agent would be able to find a schedule satisfying all the constraints. In the sequential agent scheduling case, this might not be possible.

We use *maximum matching* to tell whether a given agent is able to find a sequential schedule for all tasks $t \in T_i$ with the constraints $C(t)$ given to it. If the (polynomial) maximum matching algorithm is not able to find a complete matching for T_i, i.e., some of the tasks could not be scheduled, there must be a *scheduling conflict* between a task t in the matching and a task t' not in the matching. Such a conflict can be resolved by adding a precedence constraint $t \prec t'$ between t and t' and calling the ISA algorithm again on the extended scheduling instance. Note that the result of such extensions of the precedence relation is twofold *(i)* the conflict between t and t' is removed and *(ii)* the global makespan $d(T)$ might be increased. This matching, extending the precedence relation, and calling the ISA algorithm is repeated until we are guaranteed that for each agent there exists at least one sequential schedule. The result is a set of constraints C guaranteeing that any schedule resulting from independently chosen schedules realizes a makespan that is at most twice as long as the optimal makespan.

We assume that the tasks can be accomplished using *preemption*. This enables an agent to complete a part of task t, then to start some other tasks, process a next part of t and so on. If this is allowed, we can easily reduce a sequential scheduling instance $\langle \{T_i\}_{i=1}^n, \prec, 1, d() \rangle$ to a sequential scheduling instance with unit durations. Hence, we can reuse the approximation algorithm to obtain a 2-approximation algorithm for autonomous scheduling of tasks with arbitrary durations.

As for our future work, we plan to formally analyze the minimal degree of flexibility that the proposed algorithms can ensure given different task structures. Furthermore, we would also like to investigate the trade-off between the degree of flexibility and the loss of the makespan efficiency. Such a study would enable us to design algorithms that are better customized to specific applications.

Don't Give Yourself Away: Cooperation Revisited

Anton Nijholt

University of Twente, Human Media Interaction
Faculty EEMCS, University of Twente, Enschede, the Netherlands.

1 Introduction

Most of our research in human-computer interaction assumes that humans and computers cooperate. And although there is research on adaptive interfaces, most of the time the user has to adapt to the interface by using rather unnatural devices, follow interaction protocols, speak clearly, etcetera. Here we explore human-computer interaction where there is not necessarily cooperation and where it may be in the interest of the user to hide his intentions or feelings.

People often hide their feelings, they often hide their thoughts, and they often hide information. People often behave differently depending on when they are alone or when others are around. People sometimes want to hide from others; they are not always in need of an audience, bystanders or partners.

People have their own interests and preferences. Depending on them, and their personality and their mood, they voluntary or involuntary give away part of themselves during interactions. People do not always want to be forthcoming. Moreover, they play roles. Implicit or explicit decisions are made about the roles they want to play, what they want to disclose, and how much effort they will make in attempting to understand a conversational partner. Also, too much interest from others in our own motivations is not appreciated. We don't want other people to read our mind.

It is not always in our interest to be cooperative. Being cooperative, just as being polite, can sometimes help us to get closer to an interactional goal. We can flatter our conversational partner, we can purposely misunderstand our partner in order to make a humorous remark, and we can play the devil's advocate, and nevertheless be cooperative. We play along with the rules of a conversation or negotiation game and therefore we are cooperative despite possible elements of competitiveness. In these situations Grice's maxims on cooperation, i.e. assumptions a listener is supposed to have about the interaction behaviour of a speaker, seem to be violated, but the relevance of the behaviour can be explained from a pragmatic, conversational point of view, rather than from a sentence level point of view. Conversational partners can achieve their goals although they can have different interests.

Obviously, there is not necessarily a balance between capabilities of conversational partners. Partners differ in background, knowledge, attitudes and personality. A partner can be more determined to reach a certain goal, a partner can have more social intelligence and be able to read the mind of its human opponent better than he or she is able to do.

2 Disappearing Computers and Interfaces

Interface technologies now include speech and language input, haptic input, and vision input. Home and recreational computer use is important and requires interfaces where there is a user access layer where user friendliness, ease of learning, adaptiveness, and fun to use are main design issues. But there is more. Since we can have sensors embedded in the environment, including walls, furniture, devices, robots and pets, the environment has become intelligent and it can perform not only reactive, but also pro-active behaviour, trying to anticipate what the inhabitant is doing and doing this by perceiving activities and verbal and nonverbal behaviour. Embedded sensors include cameras, microphones, location and movement sensors, and sensors

that collect and distinguish various types of physiological information and brain activity patterns. Information about the behaviour of the inhabitants and their implicit and explicit addressing of the environment can be fused and interpreted in order to support the inhabitants. In these environments humanoids and pet-like devices can play a useful role in observing inhabitants and interacting with them. Agent-modelled virtual humans can have specific tasks in the sensor-equipped environments (e.g., be a friend, assist in cooking, take care of house security, assist in health-care and fitness, be an opponent in games or sports), they can represent human beings (e.g., family members that are away from home), and they can communicate with each other, distributing their knowledge about the environments' real and virtual inhabitants.

3 Not Giving Away Your Intentions or Feelings

There is the underlying assumption that we are acting and behaving in a smart environment that is inhabited by agents that perceive your acting and behaviour and that may profit from the knowledge that is obtained in that way. And, they may have goals that do not necessarily match your interests. Agents may collect information that threatens your interests and provide that information to others. Where can we hide from the simulated social intelligence of our environments?

However, these are not the main issues we discuss. Rather we look at our behaviour during natural interactions and our reasons to hide information, i.e., not to be forthcoming or, even, aiming to mislead our (artificial) interaction partner. In future ambient intelligence environments, are we still able to provide our conversational partners with incomplete and sometimes wrong information about ourselves, our intentions and our feelings just as we are able to do and are used to do, in real-life situations nowadays with, among others, the aim to support smooth and effective conversation and interaction?

In our research on continuous interaction modelling (rather than 'turn-taking' interaction) we designed and implemented applications where it turned out, that is, not necessarily designed that way, that in these applications their users (or conversational partners) felt there sometimes were advantages in not displaying their intentions and feelings. This became clear in our research on a so-called Sensitive Artificial Listener (SAL), developed in the framework of an EU FP6 Network of Excellence on the role of emotions in the interface, in which we participated, in our research on an interactive virtual dancer, an interactive virtual conductor, and an interactive virtual fitness trainer. In these applications both the nonverbal interaction behaviour and the fact that during interactions all conversational partners continuously display nonverbal interaction behaviour, made clear that continuously decisions are being made about what you would like to become displayed to your interactional partner. Examples that emerged in our applications are: using humour to temporarily mislead your conversational partner, not being sincere by feigning interest in a conversation, not yet wanting to show your fatigue to your fitness trainer or colleagues, and feigning movements in virtual reality entertainment game.

4 Conclusions

In the full paper [1] we discuss natural situations for cooperative and non-cooperative behaviour. In particular we discuss the friction when on the one hand our smart environments and processing technologies not only allow, but also invite natural interaction behaviour, while on the other hand the processing technologies are able to extract more information about our intentions and feelings from this natural interaction behaviour than we would like to become known. How to deal with partners that have not necessarily been designed to help us, how to deal with partners, e.g. in games and sports that are opponents rather than friends? Preliminary ideas on these topics will are illustrated with examples of our research on nonverbal interaction with virtual humans.

References

[1] A. Nijholt. Don't Give Yourself Away: Cooperation Revisited. *Proc. Symp. Logic and the Simulation of Interaction and Reasoning at the AISB 2008 Convention Communication*, Interaction and Social Intelligence, 3-4 April 2008, Aberdeen, UK, 41-46.

Audiovisual Laughter Detection Based on Temporal Features

Stavros Petridis [a] Maja Pantic [ab]

[a] *Dept. Computing, Imperial College London,180 Queen's Gate, SW7 2AZ, London, UK*
[b] *EEMCS, Univ. Twente, Drienerlolaan 5, 7522 NB, Enschede, NL*

Abstract

Previous research on automatic laughter detection has mainly been focused on audio-based detection. In this study we present an audiovisual approach to distinguishing laughter from speech based on temporal features and we show that the integration of audio and visual information leads to improved performance over single-modal approaches. Static features are extracted on an audio/video frame basis and then combined with temporal features extracted over a temporal window, describing the evolution of static features over time. When tested on 96 audiovisual sequences, depicting spontaneously displayed (as opposed to posed) laughter and speech episodes, in a person independent way the proposed audiovisual approach achieves an F1 rate of over 89%.

1 Introduction

One of the most important non-linguistic vocalizations is laughter, which is reported to be the most frequently annotated non-verbal behaviour in meeting corpora. Laughter is a powerful affective and social signal since people very often express their emotion and regulate conversations by laughing. In human - computer interaction (HCI), automatic detection of laughter can be used as a useful cue for detecting the user's affective state and, in turn, facilitate affect-sensitive human-computer interfaces. Also, semantically meaningful events in meetings such as topic change or jokes can be identified with the help of a laughter detector. In addition, such a detector can be used to recognize segments of non-speech in automatic speech recognition and for content-based video retrieval.

Few works have been recently reported on automatic laughter detection. The main characteristic of the majority of these studies is that only audio information is used, i.e., visual information carried by facial expressions of the observed person is ignored. Here we present an audiovisual approach in which audio and visual features are extracted from the audio and video channels respectively and fused on decision- or feature-level fusion. The aim of this approach is to discriminate laughter episodes from speech episodes based on temporal features, i.e. features which describe the evolution of static features over time.

2 System Overview

As an audiovisual approach to laughter detection is investigated in this study, information is extracted simultaneously from the audio and visual channels. For each channel two types of features are computed: static and temporal. The static features used are the PLP coefficients for audio and 4 shape parameters for video computed in each audio/video frame respectively. The shape parameters are computed by a point distribution model, learnt from the dataset at hand, with the aim of decoupling the head movement from the movement produced by the displayed facial expressions [1]. The 4 shape parameters used are those which correspond to the facial expressions. The temporal features considered are simple statistical features, e.g. mean, standard deviation, etc, calculated over a window T together with the coefficients of a quadratic polynomial fitted in the same window T. When considering temporal features, which describe the evolution of static features over time T (size of the used temporal window), it is common to apply the same set of

Type of Fusion	Audio features	Visual Features	F1
Static Features			
Audio only	PLP + ΔPLP	-	68.18
Video Only	-	4 Shape Param.	83.49
Decision Level	PLP + ΔPLP	4 Shape Param.	86.53
Feature Level	PLP + ΔPLP	4 Shape Param.	83.72
Static Features + Temporal Features			
Decision Level	PLP + ΔPLP + AdaBoost	4 Shape Param. + Quadratic Fitting	89.31
Feature Level	PLP + ΔPLP + AdaBoost	4 Shape Param. + Quadratic Fitting	89.08

Table 1: F1 measure for the two different types of audiovisual fusion, decision and feature level fusion

functions to all static features. In other words, the assumption is made that the evolution of all static features in time can be described in the same way. However, this is not always true and it is reasonable to believe that the temporal evolution of (some) static feature(s) will be different. In order to capture those different characteristics we consider a pool of features, which contains all the temporal features. Then AdaBoost is applied (as a feature selector) to select the temporal features that best describe the evolution of each static feature.

Once the static and temporal features are extracted for both modalities, then they are fused with the two commonly used fusion methods, decision- and feature- level fusion. Neural networks are used as classifiers for both types of fusion.

3 Dataset

Posed expressions may differ in visual appearance, audio profile, and timing from spontaneously occurring behavior. Evidence supporting this hypothesis is provided by the significant degradation in performance of tools trained and tested on posed expressions when applied to spontaneous expressions. This is the reason, we use only spontaneous (as opposed to posed) displays of laughter and speech episodes from the audiovisual recordings of the AMI meeting corpus [2] in a person-independent way which makes the task of laughter detection even more challenging.In total, we used 40 audio-visual laughter segments, 5 per person, and 56 audio-visual speech segments.

4 Results

We compare the performance of different temporal features for both single-modal and audiovisual detectors. Our results show that each static feature is best described in time by the combination of several temporal features (which are different for each static feature) rather than a fixed set of temporal features applied to all static features. It has been also demonstrated that the additional information provided by the temporal features is beneficial for this task. Regarding the level at which multimodal data fusion should be performed, both decision- and feature-level fusion approaches resulted in equivalent performances when temporal features were used. However, when static features were used, decision-level fusion outperformed feature-level fusion. Our results also show that audiovisual laughter detection outperforms single-modal (audio / video only) laughter detection, attaining an F1 rate of over 89% (see Table 1).

References

[1] D. Gonzalez-Jimenez and J. L. Alba-Castro. Toward pose-invariant 2-d face recognition through point distribution models and facial symmetry. *IEEE Trans. Inform. Forensics and Security*, 2(3):413–429, 2007.

[2] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, and V. Karaiskos. The ami meeting corpus. In *Int'l. Conf. on Methods and Techniques in Behavioral Research*, pages 137–140, 2005.

$\mathrm{P^3C}$: A New Algorithm for the Simple Temporal Problem*

Léon Planken [a] Roman van der Krogt [b] Mathijs de Weerdt [a]

[a] *Faculty EEMCS, Delft University of Technology, The Netherlands*
[b] *Cork Constraint Computation Centre, University College Cork, Ireland*

The Simple Temporal Problem (STP) was first proposed by Dechter et al. [1]. An STP instance consists of a set of time-point variables and a set of constraints that bound the time difference between pairs of these variables. The goal is then to find out all possible time values that can be assigned to these variables, or the message that no such assignment exists. The STP has received widespread attention that still lasts today, with applications in diverse areas. Moreover, the STP appears as a pivotal subproblem in more expressive formalisms for temporal reasoning, such as the Temporal Constraint Satisfaction Problem—proposed by Dechter et al. in conjunction with the STP.

An STP instance $\mathcal{S}$ consists of a set $V = \{v_1, \ldots, v_n\}$ of time-point variables representing events, and a set E of m constraints over pairs of time points, bounding the time difference between events. Every constraint $c_{i\to j}$ involves two variables, v_i and v_j and has a weight $w_{i\to j} \in \mathbb{R}$ corresponding to an upper bound on the time difference between the events involved, thus representing an inequality $v_j - v_i \leq w_{i\to j}$. The STP can be conveniently represented as a constraint graph. A *solution* to the STP instance is an assignment of a real value to each time-point variable such that the differences between each constrained pair of variables fall within the range specified by the constraint. Note that many solutions may exist; to capture all of them, we are interested in calculating an equivalent *decomposable* STP instance, from which all solutions can then be extracted in a backtrack-free manner.

The fastest known algorithm to attain decomposability is $\triangle\mathrm{STP}$ [3], included as Algorithm 1. Instead of considering a complete constraint graph, as its predecessors do, this algorithm works on a *chordal* constraint graph, which generally contains far less edges. A graph is chordal if every cycle of length greater than 3 contains a *chord*, i.e. an edge connecting two nonadjacent vertices. Chordality can efficiently be enforced in $\mathcal{O}(nm)$ time [2]. Further, it is known in graph theory that a graph is chordal if and only if its vertices can be ordered in what is called a *simplicial elimination ordering*. For details, we refer to our full paper; however, it should be noted that this ordering is a byproduct of enforcing chordality.

For our new algorithm $\mathrm{P^3C}$, included as Algorithm 2, we assume the vertices are ordered in this way. To enforce decomposability, the algorithm then performs just a forward and a backward sweep along the triangles in the graph, whereas $\triangle\mathrm{STP}$ considers each triangle at least once. By this consideration, $\mathrm{P^3C}$ will never be slower than $\triangle\mathrm{STP}$. Moreover, we constructed a class $\mathcal{P}$ of pathological STP instances for which we proved the following result:

Theorem 1. *When solving an instance from $\mathcal{P}$ on t triangles, the $\triangle\mathrm{STP}$ algorithm may require processing $\Omega\left(t^2\right)$ triangles.*

In contrast, $\mathrm{P^3C}$'s runtime can be strictly bounded:

Theorem 2. *Algorithm $\mathrm{P^3C}$ achieves decomposability in time $\Theta(t)$, where t is the number of triangles in the (chordal) STP. If the instance is inconsistent, this is discovered in time $\mathcal{O}(t)$.*

Upper bounds on the time complexity can also be expressed in other parameters:

Corollary 3. *The $\mathrm{P^3C}$ algorithm has time complexity $\mathcal{O}\left(n\delta^2\right) \subseteq \mathcal{O}\left(n^3\right)$. Here, δ is the graph degree (i.e. the maximum number of neighbours of a single vertex).*

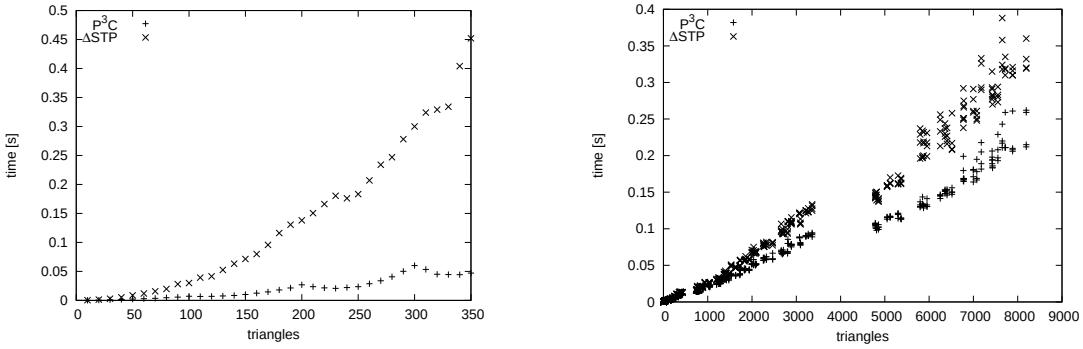
*Published in: Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), AAAI Press, 2008

$Q \leftarrow$ all triangles in $\mathcal{S}$
while $\exists T \in Q$ **do**
  foreach permutation (v_i, v_j, v_k) of T **do**
    $w_{i\to k} \leftarrow \min(w_{i\to k}, w_{i\to j} + w_{j\to k})$
    if $w_{i\to k}$ has changed **then**
      if $w_{i\to k} + w_{k\to i} < 0$ **then**
        return INCONSISTENT
      end
      $Q \leftarrow Q \cup \{\text{all triangles } \hat{T} \text{ in } \mathcal{S} \mid v_i, v_k \in \hat{T}\}$
    end
  end
  $Q \leftarrow Q \setminus \{T\}$
end

Algorithm 1: $\triangle$STP

for $k \leftarrow n$ **to** 1 **do**
  forall $i, j < k$ **such that** $\{v_i, v_k\}, \{v_j, v_k\} \in E$ **do**
    $w_{i\to j} \leftarrow \min(w_{i\to j}, w_{i\to k} + w_{k\to j})$
    if $w_{i\to j} + w_{j\to i} < 0$ **then**
      return INCONSISTENT
    end
  end
end
for $k \leftarrow 1$ **to** n **do**
  forall $i, j < k$ **such that** $\{v_i, v_k\}, \{v_j, v_k\} \in E$ **do**
    $w_{i\to k} \leftarrow \min(w_{i\to k}, w_{i\to j} + w_{j\to k})$
    $w_{k\to j} \leftarrow \min(w_{k\to j}, w_{k\to i} + w_{i\to j})$
  end
end

Algorithm 2: $\mathrm{P^3C}$



(a) Pathological instances

(b) Job-shop problem

Figure 1: Test results

We empirically validated these results with tests on the specially constructed pathological STP instances as well as more practical test cases. Some results are included in Figure 1.

References

[1] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1–3):61–95, 1991.

[2] Uffe Kjærulff. Triangulation of graphs - algorithms giving small total state space. Technical report, Aalborg University, March 1990.

[3] Lin Xu and Berthe Y. Choueiry. A new efficient algorithm for solving the Simple Temporal Problem. In *Proceedings of TIME-ICTL*, pages 210–220, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

OperA and Brahms: a symphony?[1]
Integrating Organizational and Emergent Views on Agent-Based Modeling

Bart-Jan van Putten[1,2], Virginia Dignum[1], Maarten Sierhuis[2,3], Shawn R. Wolfe[4]

[1] *Utrecht University, Dept. Information and Computing Sciences,*
Email: bnaic@bart-jan.nl / virginia@cs.uu.nl
[2] *RIACS / NASA Ames Research Center,*
[3]*Delft University of Technology, Man-Machine Interaction Group*
Email: Maarten.Sierhuis-1@nasa.gov
[4] *NASA Ames Research Center, Intelligent Systems Division,*
Email: Shawn.R.Wolfe@nasa.gov

1 Introduction

Organizations are intentionally formed to accomplish a set of common objectives, defined by its stakeholders and policy makers. The people that work for those organizations often only partially pursue those global objectives and requirements. Workers will pursue their own individual objectives as well, frequently resulting in a gap between the a priori designed flows of tasks and procedures reflecting the ideal activity of the organization (i.e., the work process), and the activities that actually get things done (i.e., the work practice) [1]. This gap does not exist only because of the difference in objectives between individuals and the organization, but also because many policy makers abstract from work practice when they design work systems (i.e., business operations). For example, it is uncommon for a job description to include 'socialize with co-workers', 'drink coffee', or 'read e-mail'.

In agent-based modeling and simulation (ABMS), the *organizational view* can be used to prescribe the desired outcomes of work processes in an organization. The *emergent view* can be used to describe the actual behavior of people in the organization, called work practice.

Policy makers can be supported in developing and evaluating operational concepts by simulating the *normative* gap between prescriptions of work processes and descriptions of work practice. Thereto, a Work Systems Modeling and Simulation framework (WSMS) has been developed, which integrates the organizational and emergent views on agent-based modeling. The WSMS is based on the integration of two independent, existing frameworks: OperA [2] and Brahms [4]. To verify and validate the WSMS, the case of Collaborative Traffic Flow Management in the United States' National Airspace System was used. A scenario from this case was simulated using the WSMS. The results show that OperA and Brahms have been integrated successfully. The output of the simulation shows which organizational objectives were (not) met and which organizational policies were (not) violated. This way, policy makers can determine to what extent the actual work practice differs from the desired work process.

2 Work Systems Modeling and Simulation framework

The design and evaluation of work systems can be supported with agent-based modeling and simulation that incorporates both an organizational view (top-down) and an emergent view (bottom-up) on work practice. Most current modeling and simulation frameworks only focus on either one of these views. In this paper we show how two multi-agent modeling frameworks, OperA, a methodology developed to represent and analyze organizational systems, and Brahms, a language developed to describe work

[1] Published in the Proceedings of the 9th International Workshop on Agent Oriented Software Engineering, AOSE@AAMAS'08, May 2008, Estoril, Portugal

practice, have been integrated. The framework allows the modeling of both the organizational objectives and the emerging (possibly divergent) work practice. By running simulations using the integrated model, it is possible to determine to what degree the workers achieve the organizational objectives. The results of these simulations are used by both policy makers and workers themselves, to understand, test, and improve work practice.

Based on the complementary viewpoints of OperA and Brahms, we hypothesized that, after integration, the two frameworks could complement each other in the following two ways: (1) OperA adds the top-down (organizational) view to Brahms, Brahms adds the bottom-up (emergent) view to OperA, so that both perspectives are represented, (2) simulations can be run that show the gap between the two perspectives. In order to realize point 2, it is necessary to first convert the OperA model to Brahms, and then to implement the actual work practice. This is done by filling in the specific behavior of the agents, which were treated as 'black boxes' in the OperA model. This results in a model that is completely described in Brahms, represents both the organizational view and the emergent view, and which is executable for simulation. In this paper we present a mapping of OperA to Brahms that meets these requirements. Approaches such as S-Moise+ [3], RNS2 [5], and [7] are similar to this research as they aim to develop organizational models to support different levels of coordination and autonomy. However, the difference is that they aim to develop open, heterogeneous multi-agent systems from an engineering perspective, whereas we aim to develop more realistic models of work practice from a human-centered perspective. The second way in which we differ from these approaches is that we provide means to populate the organization with agents specified in some agent language.

3 Conclusions

The design and evaluation of work systems can be supported with agent-based modeling and simulation that incorporates both an organizational view (top-down) and an emergent view (bottom-up) on work practice. We have integrated two frameworks, each representing one of the two views. The integrated method makes it possible to simulate work practice, and to monitor the gap between the emergent behavior and the desired outcomes as defined by the organization's policy makers.

This work contributes to our more general research objective: How can we improve models of work practice by incorporating the organizational view? What happens when agents become aware of the fact that they are violating a norm?, What is the influence of norms on work practice?, and: How do norms arise from work practice? New insights in these areas will lead to more realistic models of work practice, and thereby to improved agent-oriented system engineering methodologies.

Acknowledgements: This material is based upon work supported by NASA under award NNA07BB97C. The research of V. Dignum is funded by the Netherlands Organization for Scientific Research (NWO), through Veni-grant 639.021.509.

4 References

1. Clancey, W.J., Sachs, P., Sierhuis, M., Van Hoof, R.: Brahms: Simulating Practice for Work Systems Design, *Int. Journal on Human-Computer Studies* ***49***, 831--865 (1998)
2. Dignum, V., Dignum, F., Meyer, J. J.: An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations. *Knowledge Engineering Review* ***19****(2)*, pp. 147--174 (2004)
3. Hübner, J., Sichman, J., Boissier O.: S-MOISE+: A middleware for developing organised multi-agent systems. In: Boissier, O. et al. (eds.): *Coordination, Organizations, Institutions and Norms in MAS (COIN I)*, LNCS 3913, Springer (2006)
4. Sierhuis, M., Clancey, W.J., Van Hoof, R.: Brahms: A Multiagent Modeling Environment for Simulating Work Processes and Practices. *Int. Journal of Simulation and Process Modeling* ***3****(3)*, 134--152 (2007)
5. Van der Vecht, B., Dignum, F., Meyer, J.-J.Ch., Neef, M.: A Dynamic Coordination Mechanism Using Adjustable Autonomy. In: COIN@MALLOW'07, Durham, UK, (2007)
6. Weiss, G., Nickles, M., Rovatsos M., Fischer, F.: Specifying the intertwining of coordination and autonomy in agent-based systems. *Int. Journal Network and Computer Applications* ***29***, (2006)

Monitoring and Reputation Mechanisms for Service Level Agreements*

Omer Rana[a] Martijn Warnier[b] Thomas B. Quillinan[b]
Frances M.T. Brazier[b]

[a] *School of Computer Science/Welsh eScience Centre, Cardiff University, UK.*
[b] *Department of Computer Science, VU University, Amsterdam, The Netherlands.*

Abstract

A Service Level Agreement (SLA) is an electronic contract between a service user and a provider, and specifies the service to be provided, Quality of Service (QoS) properties that must be maintained by a provider during service provision (generally defined as a set of Service Level Objectives (SLOs)), and a set of penalty clauses specifying what happens when service providers fail to deliver the QoS agreed. Although significant work exists on how SLOs may be specified and monitored, not much work has focused on actually identifying how SLOs may be impacted by the choice of specific penalty clauses. A trusted mediator may be used to resolve conflicts between the parties involved. The objectives of this work are to: (i) identify classes of penalty clauses that can be associated with an SLA; (ii) define how to specify penalties in an extension of WS-Agreement; and (iii) specify to what extent penalty clauses can be enforced based on monitoring of an SLA.

A Service Level Agreement (SLA) is an agreement between a client and a provider in the context of a particular service provision. SLAs may be between two parties, for instance, a single client and a single provider, or between multiple parties, for example, a single client and multiple providers. SLAs specify Quality of Service (QoS) properties that must be maintained by a provider during service provision – generally defined as a set of Service Level Objectives (SLOs). Often an SLA is only relevant when a client *directly* invokes a service (rather than through an intermediary – such as a broker). Such direct interaction also implies that the SLOs need to be measurable, and must be monitored during the provision of the service.

From an economics perspective, one may associate a cost with an SLA – that is the amount of money a client needs to pay the provider if the agreement has been adhered to. Adhering to an agreement entails that the requested quality has been met. The cost needs to be agreed between a client and a provider – and may be based on a posted price (provider publishes), or negotiated through single/multi-round auctions (for example, English, Dutch, or Double). How this price is set has been considered elsewhere [1]. However, the mechanism for setting the price can also be determined through equilibrium pricing (based on supply-demand) or through auctions (based on client need). An SLA must also contain a set of penalty clauses specifying the implications of failing to deliver the pre-agreed quality. This penalty may also be defined as a cost – implying that the total revenue made by a provider would be the difference between the cost paid by the client and the discount (penalty) imposed on the provider. This type of analysis assumes that failure to meet an SLA is a non-binary decision – that is, an SLA may be "partially" violated, and that some mechanism is in place to determine how this can be measured.

Although significant work exists on how SLOs may be specified and monitored [5], not much work has focused on actually identifying how SLOs may be impacted by the choice of specific penalty clauses. A trusted mediator may be necessary to resolve conflicts between involved parties. The outcome of conflict resolution depends on the situation: penalties, impact on potential future agreements between the parties and the mandatory re-running of the agreed service, are examples. While it may seem reasonable to penalise SLA non-compliance, there are a number of concerns when issuing such penalties. For example, determining whether the service provider is the only party that should be penalised, or determining the type of penalty that is applied to each party. Enforcement in the various legal systems of different countries can be tackled

*The full version [6] of this paper appeared in the Proceedings of the Fifth International Workshop on Grid Economics and Business Models (GECON 2008).

through stipulating a 'choice of law clause', that is, a clause indicating expressly the country whose laws will be applied in case a conflict between the provider and the client would occur.

Automating the conflict resolution process could provide substantial benefits. Broadly speaking there are two main approaches for contractual penalties in SLAs: reputation based mechanisms [4, 7] and monetary fines. It is useful to note that often obligations within an SLA are primarily centered on the provider towards the client. An SLA is, therefore, an agreement between the provider to offer particular QoS to a client for some monetary return. We do not consider scenarios where the client also has an obligation towards the provider. An example of such a scenario could be where a provider requires the client to make input data available by a certain time frame to ensure that a particular execution time target is met. If the client is unable to meet the deadline for making such data available, the penalty incurred by the provider would no longer apply.

The use of reputation-based mechanisms to promote data integrity in distributed architectures has been explored by [3]. Knowing the reputation of a client can provide insight into what access may be granted to that client by a provider. Maintaining a measure of each client's reputation allows clients to make decisions regarding the best service provider for a specific task. In this case, reputation is a numerical value quantifying compliance to one or more SLAs. This value represents the previous behaviour of the provider in the system, and can be used by other clients to determine whether or not to interact with that provider. The higher this value, the greater the belief that the provider will act correctly in the future. Applying a numerical weight to users allows a more informed decision to be made when negotiating SLAs in the future. As users (clients and providers) interact with one another in the system, their reputation changes to reflect how they perform. For example, if a service provider consistently provides poor service (that is, violating its SLAs), its reputation will decline.

While reputation based mechanisms work relatively well in community based environments – where each participant monitors and judges other participants – in commercial environments reputation based mechanisms are rarely used. This can partly be attributed to the unbalanced nature of the relationship between clients and service providers. Monetary fines give a higher degree of expected QoS for service providers and (especially) clients. Such approaches are not new, other works in this area, such as [2], provide only a partial solution to this problem. For example, they do not have a mechanism for conflict resolution.

References

[1] M. Becker, N. Borrisov, V. Deora, O. F. Rana, D. Neumann, "Using k-Pricing for Penalty Calculation in Grid Market", *Proceedings of IEEE HICSS 2008 Conference, Hawaii, January 2008*.

[2] B. C. Clayton, T. B. Quillinan, and S. N. Foley. Automating security configuration for the Grid. In *Journal of Scientific Programming*, 13(2):113–125, 2005.

[3] A. Gilbert, A. Abraham, and M. Paprzycki. A System for Ensuring Data Integrity in Grid Environments. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, pages 435–439, Las Vegas, Nevada, USA, April 5–7 2004.

[4] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th Int. World Wide Web Conference*, Budapest, Hungary, May 20-24 2003. ACM Press.

[5] A. Keller and H. Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management*, 11(1):57–81, 2003.

[6] O. Rana, M. Warnier, T.B. Quillinan and F.M.T. Brazier Monitoring and Reputation Mechanisms for Service Level Agreements., In *Proceedings of the 5th International Workshop on Grid Economics and Business Models (GenCon)*, Las Palmas, Gran Canaria, Spain, August, 2008. Springer Verlag.

[7] J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44–56, 2002.

Subjective Machine Classifiers

Dennis Reidsma Rieks op den Akker

Human Media Interaction, University of Twente, PO Box 217
NL-7500 AE, Enschede, The Netherlands

Abstract

Many interesting phenomena in conversations require interpretative judgements by the annotators. This leads to data which is annotated with lower levels of agreement due to the differences in how annotators interpret conversations. Instead of throwing away this data we show how and when we can exploit it. We analyse the (dis)agreements between annotators for two different cases in a multimodal annotated corpus and explicitly relate the results to the way machine-learning algorithms perform on the annotated data.

Exploiting Subjective Annotated Data

Human classifications of events in terms of rather intuitive notions ask from the annotator to interpret social aspects of human behavior, such as the speaker's intention expressed in a conversation. We argue that dis-agreements between different observers is unavoidable and an intrinsic quality of the interpretation and classification process of such type of content. Any sub-division of these type of phenomena into a pre-defined set of disjunct classes suffers from being arbitrary. There are always cases that can belong to this but also to that class. Analysis of annotations of the same data by different annotators may reveal that there are differences in the decisions they make. These difference may reveal some personal preference for one class over another. (For references see the full paper [4].)

Instead of throwing away the data as not being valuable at all for machine learning purposes, we show two ways to exploit such data, both leading to high precision / low recall classifiers that in some cases refuse to give a judgement. The first way is based on the identification of subsets of the data that show higher inter-annotator agreement. When the events in these subsets can be identified computationally the way is open to use classifiers trained on these subsets. We illustrate this with several subsets of addressing events in the AMI meeting corpus and we show that this leads to an improvement in the accuracy of the classifiers. Precision is raised in case the classifier refrains from making a decision in those situation that fall outside the subsets. The second way is to train a number of classifiers, one for each of the annotators data part of the corpus, and build a Voting Classifier that only makes a decision in case all classifiers agree on the class label. This approach is illustrated by the problem of classification of the dialogue act type of Yeah-utterances in the AMI corpus. The results show that the approach indeed leads to the expected improvement in precision, at the cost of a lower recall, because of the cases in which the classifier doesn't make a decision.

Some types of disagreement are more structural and other types are more noise like. We focus on a way of coping with disagreements resulting from a low level of intersubjectivity that actively exploits the structural differences in the annotations caused by this. From the patterns in the disagreements between annotators, we are able to formulate constraints and restrictions on the use of the data and on the reliability of the classifier's judgements. (see also Reidsma and Carletta [3]).

To illustrate how these ideas work out we used the hand annotated face-to-face conversations from the 100 hour multi-modal AMI meeting corpus [1]. A part of this corpus is annotated (by three annotators) with addressee information. Real dialogue acts were assigned a label indicating who the speaker is talking to. In these type of meetings most of the time the speaker addresses the whole group, but sometimes his dialogue act is particularly addressed to some individual (about 2743 of the 6590 annotated real dialogue acts); for example because he wants to know that individual's opinion. DAs are either addressed to the group (*G-addressed*) or to an individual (*I-addressed*). Another layer of the corpus contains *focus of attention*

information. so that for any moment it is known whether a person is looking at the table, white board, or some other participant.

The level of agreement with which an utterance is annotated with addressee is dependent on the FOA context of an utterance. We expect this will be reflected directly by the machine learning performance in these two contexts: the low agreement might indicate a context where addressee is inherently difficult to determine and furthermore the context with high agreement will result in annotations containing more consistent information that machine learning can model.

To verify this assumption we experimented with automatic detection of the addressee of an utterance based on lexical and multimodal features. Roughly 1 out of every 3 utterances is performed in a context where the speaker's FOA is not directed at any other participant. This gives us three contexts to train and to test on: all utterances, all utterances where the speaker's FOA is not directed at any other participant (1/3 of the data) and all utterances during which the speaker's FOA is directed at least once at another participant (2/3 of the data). The outcome shows a performance gain in contexts with a distinctive addressee-directed focus of attention of the speaker.

We can expect that a classifier A trained on data annotated by A will perform better when tested on data annotated by A, than when tested on data annotated by B. In other words, classifier A is geared towards modelling the 'mental conception' of annotator A. Suppose that we build a Voting Classifier, based on the votes of a number of classifiers each trained on a different annotator's data. The Voting Classifier only makes a decision when all voters agree on the class label. How good will the Voting Classifier perform? Is there any relation between the agreement of the voters, and the agreement of the annotators? Will the resulting Voting Classifier in some way embody the overlap between the 'mental conceptions' of the different annotators?

As an illustration and a test case for such a Voting Classifier, we consider the human annotations and automatic classification of *"Yeah-utterances"*, utterances that start with the word "yeah". They make up about eight percent of the dialogue acts in the AMI meeting conversations. In order to get information about the stance that participants take with respect towards the issue discussed it is important to be able to tell utterances of "Yeah" as a mere backchannel, from Yeah utterances that express agreement with the opinion of the speaker (see the work of Heylen and Op den Akker [2]).

The class variables for dialogue act types of Yeah utterances that are distinguished are: Assess (as), Backchannel (bc), Inform (in), and Other (ot). For each annotator, a disjunct train and test set have been defined. The inter-annotator agreement on the Yeah utterances is low (pair-wise alpha values are around 0.4).

We train three classifiers DH, S9 and VK, each trained on train data taken from one single annotator, and we build a Voting Classifier that outputs a class label when all three 'voters' give the same label; the label 'unknown' otherwise. As was to be expected, the *accuracy* for this Voting Classifier is much lower than the accuracy of each of the single voters and than the accuracy of a classifier trained on a mix of data from all annotators due to the many times the Voting Classifier assigns the label 'unknown' which is not present in the test data and is always false. The precision of the Voting Classifier however is higher than that of any of the other classifiers, for each of the classes.

References

[1] Jean C. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, May 2007.

[2] D. Heylen and H.J.A. op den Akker. Computing backchannel distributions in multi-party conversations. In J. Cassell and D. Heylen, editors, *Proceedings of the ACL Workshop on Embodied Language Processing, Prague*, volume W07-19, pages 17–24, Prague, Czech Republic, June 2007. Association of Computational Linguistics.

[3] Dennis Reidsma and Jean C. Carletta. Reliability measurement without limits. *Computational Linguistics*, 2008. to appear.

[4] Dennis Reidsma and H.J.A. op den Akker. Exploiting 'subjective' annotations. In Ron Artstein, Gemma Boleda, Frank Keller, and Sabine Schulte im Walde, editors, *Proceedings of the Coling Workshop on Human Judgments in Computational Linguistics*, August 2008. to appear.

Single-Player Monte-Carlo Tree Search[1]

Maarten P.D. Schadd Mark H.M. Winands H. Jaap van den Herik
Guillaume M.J-B. Chaslot Jos W.H.M. Uiterwijk

Games and AI Group, MICC, Faculty of Humanities and Sciences,
Universiteit Maastricht, Maastricht, The Netherlands
{maarten.schadd,m.winands,herik,g.chaslot,uiterwijk}@micc.unimaas.nl

Recently, Monte-Carlo (MC) methods have become a popular approach for intelligent play in games. MC simulations have first been used as an evaluation function inside a classical search tree [2, 3]. In this role, MC simulations have been applied to Backgammon [14], Clobber [11], and Phantom Go [4]. Due to the costly evaluation, the search is not able to investigate the search tree sufficiently deep in some games [2].

Therefore, the MC simulations have been placed into a tree-search context in multiple ways [5, 7, 10]. The resulting general method is called Monte-Carlo Tree Search (MCTS). It is a best-first search where the MC simulations guide the search. MCTS builds a search tree employing Monte-Carlo evaluations at the leaf nodes. Each node in the tree represents an actual board position and typically stores the average score found in the corresponding subtree and the number of visits. In general, MCTS consists of four steps, repeated until time has run out [6]. (1) A *selection strategy* is used for traversing the tree from the root to a leaf. (2) A *simulation strategy* is used to finish the game starting from the leaf node of the search tree. (3) The *expansion strategy* is used to determine how many and which children are stored as promising leaf nodes in the tree. (4) Finally, the result of the MC evaluation is propagated backwards to the root using a *back-propagation strategy*.

Especially in the game of Go, which has a large search space [1], MCTS methods are successful [5, 7]. So far, MCTS has been applied rarely in one-player games. The only example we know of is the Sailing Domain [10]. There, it is applied on a game with uncertainty. So, to the best of our knowledge, MCTS has not been used in a one-player game with *perfect information* (a puzzle). The traditional approaches to puzzles [9] are applying A* [8] or IDA* [12]. These methods have been quite successful for solving puzzles. The disadvantage of the methods is that they need an admissible heuristic evaluation function. The construction of such a function can be difficult. Since MCTS does not need an admissible heuristic, it may be an interesting alternative. In this paper we investigate the application of MCTS to the puzzle SameGame [13] by introducing a new MCTS variant, called Single-Player Monte-Carlo Tree Search (SP-MCTS). We adapted MCTS by two modifications resulting in SP-MCTS. The modifications are (1) the selection strategy and (2) the back-propagation strategy. Moreover, SP-MCTS makes use of a straightforward Meta-Search extension.

Based on the experiments and results achieved in SameGame, we are able to provide three observations and subsequently two conclusions.

First, we observed that our TabuColorRandom play-out strategy significantly increased the score of the random simulations in SameGame. Compared to the pure random simulations, an increase of 50% in the average score is achieved.

Next, we observed that it is important to build deep SP-MCTS trees. Exploitation works better than exploration at short time controls. At longer time controls the balanced setting achieves the highest score, and the exploration setting works better than the exploitation setting. However, exploiting the local maxima still leads to comparable high scores.

Third, with respect to the extended SP-MCTS endowed with a straightforward Meta-Search, we observed that for SameGame combining a large number of small searches can be more beneficial than doing one large search.

[1]The full version of this paper is published in: *Computers and Games (CG 2008)*, LNCS 5131, pp. 1–12, Springer, Heidelberg, Germany.

The first conclusion is that SP-MCTS produced the highest score found so far for the standardized test set of 20 SameGame positions. It was able to achieve 73,998 points, breaking the old record by 1,182 points. A second conclusion is that we have shown that SP-MCTS is applicable to a one-person perfect-information game. SP-MCTS is able to achieve good results in SameGame. So, SP-MCTS is a worthy alternative for puzzles where a good admissible estimator cannot be found.

Acknowledgments

This work is funded by the Dutch Organisation for Scientific Research (NWO) in the framework of the project TACTICS, grant number 612.000.525.

References

[1] B. Bouzy and T. Cazanave. Computer Go: An AI-Oriented Survey. *Artificial Intelligence*, 132(1):39–103, 2001.

[2] B. Bouzy and B. Helmstetter. Monte-Carlo Go Developments. In H.J. van den Herik, H. Iida, and E.A. Heinz, editors, *Proceedings of the 10th Advances in Computer Games Conference (ACG-10)*, pages 159–174, Kluwer Academic, Dordrecht, The Netherlands, 2003.

[3] B. Brügmann. Monte Carlo Go. Technical report, Physics Department, Syracuse University, 1993.

[4] T. Cazenave and J. Borsboom. Golois Wins Phantom Go Tournament. *ICGA Journal*, 30(3):165–166, 2007.

[5] G.M.J.B. Chaslot, J-T. Saito, B. Bouzy, J.W.H.M. Uiterwijk, and H.J. van den Herik. Monte-Carlo Strategies for Computer Go. In P.Y. Schobbens, W. Vanhoof, and G. Schwanen, editors, *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence*, pages 83–91, Namur, Belgium, 2006.

[6] G.M.J.B. Chaslot, M.H.M. Winands, J.W.H.M. Uiterwijk, H.J. van den Herik, and B. Bouzy. Progressive strategies for Monte-Carlo Tree Search. In P. Wang et al., editors, *Proceedings of the 10th Joint Conference on Information Sciences (JCIS 2007)*, pages 655–661. World Scientific Publishing Co. Pte. Ltd., 2007.

[7] 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, *Proceedings of the 5th International Conference on Computer and Games*, volume 4630 of *Lecture Notes in Computer Science (LNCS)*, pages 72–83. Springer-Verlag, Heidelberg, Germany, 2007.

[8] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernatics*, SSC-4(2):100–107, 1968.

[9] G. Kendall, A. Parkes, and K. Spoerer. A Survey of NP-Complete Puzzles. *ICGA Journal*, 31(1):13–34, 2008.

[10] 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 *Lecture Notes in Computer Science (LNCS)*, pages 282–293, Berlin, 2006. Springer-Verlag, Heidelberg, Germany.

[11] L. Kocsis, C. Szepesvári, and J. Willemson. Improved Monte-Carlo Search, 2006. http://zaphod.aml.sztaki.hu/papers/cg06-ext.pdf.

[12] R.E. Korf. Depth-first iterative deepening: An optimal admissable tree search. *Artificial Intelligence*, 27(1):97–109, 1985.

[13] K. Moribe. Chain shot! *Gekkan ASCII*, (November issue), 1985. (In Japanese).

[14] G. Tesauro and G.R. Galperin. On-line policy improvement using Monte Carlo search. In M.C. Mozer, M.I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 1068–1074, MIT Press, Cambridge, 1997.

Mental State Abduction of BDI-Based Agents[1]

M.P. Sindlar M.M. Dastani F. Dignum J.-J.Ch. Meyer

Utrecht University
P.O. Box 80.089, 3508 TB, The Netherlands

1 Introduction

Intelligent computational agents implemented in BDI-inspired programming languages are suited for developing virtual (non-player) characters for computer games and simulations [2]. In this work we tackle the problem of providing explanations for the observed behavior of virtual characters implemented as BDI-based agents in terms of their mental states.

Agents in BDI-inspired programming languages such as 2APL, Jack, Jadex and Jason, are programmed in terms of mentalistic notions like beliefs, goals, and plans. In order for agents to deliberately cooperate with or obstruct the plans of other agents, it is a prerequisite that they can draw (defeasible) conclusions about those other agents' mental states, a capacity we refer to as mental state inference. Because agent languages with declarative mental states are often logic-based, and because a logical relation mental state $\Rightarrow$ behavior can (approximately) be identified, logical abduction — a way for inferring explanations for an observed fact — is a promising approach to providing explanations of agents' observed behavior in terms of mental state descriptions.

We envision the use of techniques for mental state inference as a means of designing characters that show a higher degree of believability. Agents that incorporate beliefs about the mental state of other agents into their plans can be expected to show more *socially aware* behavior. When explaining the behavior of others, they either correctly infer their goals or beliefs and act accordingly, or they arrive at incorrect conclusions based on explanations that are nonetheless plausible, which gives rise to a form of erroneous behavior the user understands and tolerates. Based on their hypotheses, agents can also form expectations with respect to other agents' future actions. Note that to actually *use* beliefs about the mental state of others, agents would need to have models of the mental states of other agents. Abduced explanations would have to be incorporated into these mental state models, requiring a revision mechanism and preservation of integrity constraints. We do not deal with this aspect yet, but separate the agent from the observer and focus solely on ways to find plausible explanations, without worrying how these explanations will be used.

2 Mental State Abduction

In this paper, we investigate an approach to inferring the mental state of BDI agents based on their observed behavior, using a simplified general language for programming BDI-based agents based on 2APL [1]. We assume that the actions of agents are observable as atomic, unambiguous facts. Not all actions are necessarily perceived, but *if* an action is perceived, this occurs as an event that is an unambiguous representation of this action. This assumption avoids the introduction of uncertainty about which action is perceived.

BDI agents operate by means of goal-directed behavioral rules, which are selected and executed by way of some deliberation process. We assume that an agent selects a plan only if it is not actively dealing with some other plan. This means that at no point an agent is executing more than one plan. Furthermore, we assume that an agent only executes a plan to achieve some goal it explicitly has. Moreover, plans may be discarded, either because the goal for which they were selected has been achieved, or has been dropped. A

[1]The full version of this paper appeared in the DALT workshop at AAMAS, May 2008, Lisbon. This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

strong restriction we make is that the observing entity, which perceives an agent's actions and attempts to infer its mental state, has knowledge of the entire set of operational rules of this agent. This assumption is justified in a game environment where all agents are designed by the same party.

In our approach to the problem of explaining the actions of agents in terms of their mental states we have drawn inspiration from work on abduction in propositional logic. Inferring the mental states of agents based on their observed behavior is what we call *mental state abduction*.

The observer relates perceived actions to a plan on grounds of partial traces that are structurally related to some full trace of a plan. This allows for incomplete observation, meaning the observer fails to perceive one or more actions. To identify these we specify the prefix, suffix, subsequence and dilution relations $\preccurlyeq$, $\succcurlyeq$, $\sqsubseteq$, and $\lesssim$, respectively, which are all partial orders on a language of observables. These relations reflect different perceptory conditions, such as the observer's assumption about the nature of the environment, or its status regarding perception of the agent's actions.

- **Complete observation:** If complete observation is assumed, then the observer expects to see every action the agent performs.
- **Late observation:** Late observation reflects the observer's assumption that it has possibly failed in seeing one or more of the initial actions the agent performed. From the moment it starts observing the observer expects to see every future action of the agent.
- **Partial observation:** In the case of partial observation, the observer assumes that it might fail to see some of the actions the agent performs. Such failure might occur due to some environmental circumstance, or due to the agent deliberately obscuring its actions.

Descriptions of the form $\mathsf{mental\ state} \Rightarrow \mathsf{behavior}$ capture a logical connection between the agent's mental state and some plan. To abduce the mental state based on those descriptions, the observer somehow has to compare the sequence of actions it has perceived to some plan it believes the agent might be executing. In the case of complete observation, the observer expects to see every action the agent performs. This means the first action the observer has perceived should be the first action of one or more of the traces of the plan the agent is executing, and of which the observer should have information in its description of the agent's rules. Following the same line of reasoning, any sequence of actions the observer perceives should be the prefix of one of those traces. Late observation presumes the observer might have missed the initial actions of some plan, but has seen any actions performed since it started observing. Therefore, any observed sequence must be a subsequence of some plan trace. Partial observation presumes the agent has seen some actions of a plan — in the same order as they occurred in the plan — but might have missed others. It will be no surprise that in this case the observed sequence is a dilution of some plan trace.

Using the above concepts the full paper formalizes the different notions of abduction under the different observation conditions. It proves that the number of possible explanations of behavior increases from complete to partial observations, while the number of explanations monotonically decreases if successive observations are made in the same trace using the same explanation type.

3 Conclusion

In this paper we presented a technique called mental state abduction, which allows for inferring the mental states of BDI-agents based on their observed behavior, using an approach inspired by abduction in propositional logic. Three explanatory strategies were presented, based on perceptory conditions reflecting either properties under the observer's influence, or properties of the environment beyond the observer's control. Our technique was shown to work for agents programmed in a simplified BDI-based programming language, as demonstrated by formal proofs of some interesting properties.

References

[1] Mehdi Dastani. 2APL: A practical agent programming language. *Autonomous Agents and Multi-Agent Systems*, 16:214–248, 2008.

[2] Emma Norling and Liz Sonenberg. Creating interactive characters with BDI agents. *Proceedings of the Australian Workshop on Interactive Entertainment*, pages 69–76, 2004.

Decentralized Performance-aware Reconfiguration of Complex Service Configurations

Sander van Splunter Pieter H.G. van Langen Frances M.T. Brazier

Intelligent Interactive Distributed Systems,
VU University Amsterdam,
The Netherlands

Execution of complex workflows is one of many applications of Peer-to-Peer (P2P) networks. Sustaining complex workflows in a dynamic P2P network requires adaptation when hosts involved in the workflow lose network connectivity. Adaptation of complex workflows without centralised monitoring and control is a challenge, and is often limited to instance replacement. If requirements are given on the performance of the complex workflow as a whole, then possibilities for automated adaptation are limited even more. This paper [1] describes how adaptation of complex web service configurations can be applied on complex workflows in P2P networks.

The paper describes the structures needed for automated reconfiguration of service compositions, and how performance awareness can be created. The structures described are: web services, templates, and configurations. For each structure, the relation to the nodes on the P2P network is explained. In Figure 1 a template-based configuration is displayed, where templates and web services are used to define a layered configuration. The template definitions contains propagation functions, to determine properties of the overall configuration, and inverse propagation functions to disseminate requirements, posed on the overall configurations, to the individual web services and templates.

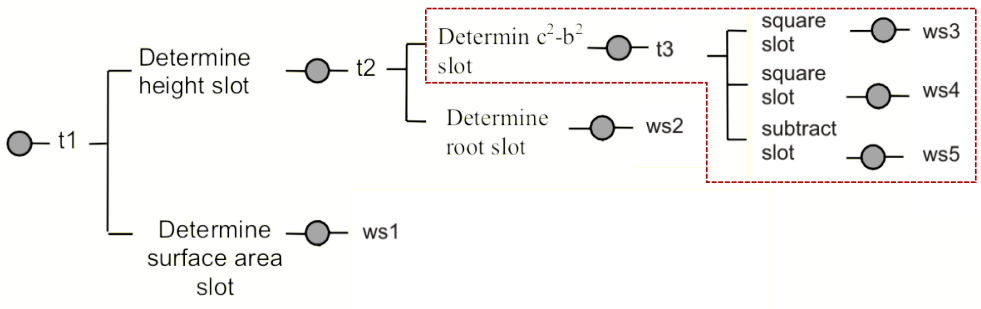


Figure 1: A template-based configuration of web services for mathematical services. The highlighted part is discussed in detail in the paper.

Figure 2 illustrates a mapping from a template-based configuration to a deployed configuration on a P2P network. The P2P communication layer is responsible for the exchange of the service and template descriptions. Over the P2P layer the template-based configuration is deployed. The figure shows the deployment of the highlighted part of the template-based configuration of Figure 1.

The paper describes a reconfiguration scenario, based on the failure of one of the services. The reconfiguration scenario involves reasoning on Quality of Service requirements posed on the overall configuration. By applying the presented approach in a P2P environment, the following options are enabled:

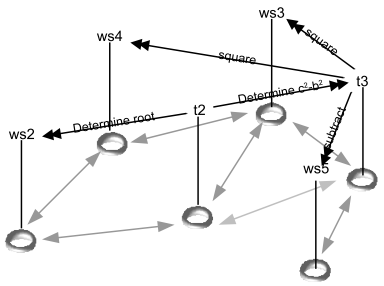


Figure 2: Part of the template-based configuration deployed over a P2P network.

- more complex adaptation beyond instance replacement,
- local and distributed reconfiguration,
- local awareness of QoS constraints on the overall configuration.

Monitoring can be enhanced, as the distribution of requirements allows local and continuous monitoring of, e.g., more dynamic performance parameters. By deploying a template-based web service configuration over the P2P network, effectively a distributed adaptive overlay network is created. The adaptivity of this overlay network can be further enhanced by clustering nodes based on dependencies as defined in the slot requirements. The property propagation functions, and the inverse propagation functions offer the ability to reason on the effect of the properties of a single service on the properties of whole overlay network.

Acknowledgments

This project is supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024 in the Interactive Collaborative Information Systems (ICIS) project (`http://www.icis.decis.nl/`), and by the NLnet Foundation (`http://www.nlnet.nl`).

References

[1] S. van Splunter, P.H.G. van Langen, and F.M.T. Brazier. Decentralized performance-aware reconfiguration of complex service configurations. In *Proceedings of the Performance for Peer-to-Peer Systems (P4P2P) workshop*. University of Warwick, May 2008.

Combined Support Vector Machines and Hidden Markov Models for Modeling Facial Action Temporal Dynamics - Abstract

Michel Valstar Maja Pantic

Department of Computing, Imperial College London,
180 Queen's gate, London SW7 2AZ, England
{*Michel.Valstar, M.Pantic*}*@imperial.ac.uk*

Abstract

The analysis of facial expression temporal dynamics is of great importance for many real-world applications. Being able to automatically analyse facial muscle actions (Action Units, AUs) in terms of recognising their neutral, onset, apex and offset phases would greatly benefit application areas as diverse as medicine, gaming and security. The base system in this paper uses Support Vector Machines (SVMs) and a set of simple geometrical features derived from automatically detected and tracked facial feature point data to segment a facial action into its temporal phases. We propose here two methods to improve on this base system in terms of classification accuracy. The first technique describes the original time-independent set of features over a period of time using polynomial parametrisation. The second technique replaces the SVM with a hybrid SVM/Hidden Markov Model (HMM) classifier to model time in the classifier. Our results show that both techniques contribute to an improved classification accuracy. Modeling the temporal dynamics by the hybrid SVM-HMM classifier attained a statistically significant increase of recall and precision by 4.5% and 7.0%, respectively.

1 Introduction

A system capable of analysing facial actions would have many applications in a wide range of disciplines. For instance, in medicine, it could be used to continuously monitor a patients pain level or anxiety, in gaming a virtual avatar could be directed to mimic the user's facial expressions and in security the analysis of facial expressions could be used to assert a person's credibility.

The method proposed in this paper is based on the analysis of atomic facial actions called Action Units (AUs), which are defined by the Facial Action Coding System (FACS) [3]. FACS is the best known and the most commonly used system developed for human observers to objectively describe facial activity in terms of visually observable facial muscle actions (AUs). FACS defines 9 upper face AUs and 18 lower face AUs. There are 5 additional AUs that belong to neither the upper nor the lower face and are often omitted in the literature.

Many of the aforementioned applications of automatic facial expression analysis require the explicit analysis of the temporal dynamics of AU activation. The body of research in cognitive sciences which suggests that the temporal dynamics of human facial behaviour (e.g. the timing and duration of facial actions) are a critical factor for interpretation of the observed behaviour, is large and growing [2, 1, 4]. In this light, it is striking that very few research groups focus on the analysis of temporal dynamics. Almost all existing facial expression recognition systems are only capable of recognising the *presence* of a facial action, regardless whether that action has just begun and is getting stronger, is at its peak or is returning to its neutral state. Also, while many systems are in essence capable to compute the total duration of a facial action based on this activation detection, none do this explicitly [7]. But perhaps more importantly, this total activation duration information alone would be insufficient for the complex tasks described above.

We chose to analyse the facial action temporal dynamics explicitly. A facial action, in our case an AU activation, can be in any one of four possible phases: (i) the onset phase, where the muscles are contracting and the appearance of the face changes as the facial action grows stronger, (ii) the apex phase, where the facial action is at its peak, (iii) the offset phase, where the muscles are relaxing and the face returns to its neutral appearance and (iv) the neutral phase, where there are no signs of this particular facial action. Often the order of these phases is neutral-onset-apex-offset-neutral, but other combinations such as multiple-apex facial actions are possible as well.

The method we propose in this paper is fully automatic, operating on videos of subjects recorded from a near-frontal view. It uses data from 20 tracked facial points to analyse facial expressions. This paper proposes two ways to improve the system presented earlier [8]. First we define a set of mid-level parameters that better encodes the dynamics of facial expressions. Instead of defining the features for each frame, we describe the evolution of the feature values over a period in time using polynomial parametrisation. This way, we capture how a feature related to a specific facial action behaves dynamically instead of observing its value at a single moment.

The second improvement we propose is to explicitly incorporate a sense of time in the classification procedure. We do so by combining a Hidden Markov Model (HMM) with a Support Vector Machine (SVM). While the evolution of a facial action in time can be efficiently represented using HMMs, the distinction between the temporal phases at a single point in time is usually made using Gaussian mixture models, which do not offer a high discrimination. SVMs on the other hand are large-margin classifiers known for their excellent discrimination performance in binary decision problems but they do not incorporate a model of time.

We have evaluated our proposed methods on 196 videos selected from the MMI-Facial Expression Database [6], containing videos of 23 different AUs. We have chosen this database, instead of for example the Cohn-Kanade DFAT-504 dataset [5], because the videos in the MMI-Facial Expression Database display the full neutral-expressive-neutral pattern. This is essential, as it is this temporal pattern of facial actions that we are interested in. The videos were chosen so that we selected at least 10 videos of every AU we want to analyse.

From our results we can conclude that replacing the SVM classifier with a hybrid SVM-HMM classifier results in a significant classification accuracy improvement. It is harder to judge the effect of adding the polynomial representation of the original mid-level parameters. Using these features, there seems to be a slight but statistically insignificant improvement, which is more pronounced in combination with the SVM-HMM classifier than it is with the original SVM classifier. However, the performance increase due to the polynomial mid-level parameters is not significant and might not be high enough to make up for the fact that we have to compute 2.5 times as many features.

References

[1] J. N. Bassili. Facial motion in the perception of faces and of emotional expression. *J. Experimental Psychology*, 4(3):373–379, 1978.

[2] J. F. Cohn and K. L. Schmidt. The timing of facial motion in posed and spontaneous smiles. *J. Wavelets, Multi-resolution and Information Processing*, 2(2):121–132, 2004.

[3] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System*. A Human Face, 2002. Salt Lake City.

[4] U. Hess and R. E. Kleck. Differentiating emotion elicited and deliberate emotional facial expressions. *European J. of Social Psychology*, 20(5):369–385, 1990.

[5] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[6] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. *Proc. Int'l Conf. Multimedia & Expo*, pages 317–321, 2005.

[7] Y. L. Tian, T. Kanade, and J. F. Cohn. *Handbook of Face Recognition*. Springer, 2005. New York.

[8] M. F. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, page 149, 2006.

Reconfiguration Management of Crisis Management Services

J.B. van Veelen[a] S. van Splunter[b] N.J.E. Wijngaards[a] F.M.T. Brazier[b]

[a] *Thales Research & Technology/D-CIS Lab Postbus 90 Delft*
[b] *Vrije Universiteit Amsterdam, Boelelaan 1081, 1081HV Amsterdam*

1 Extended abstract

IT-support for adaptation of complex systems is widely available if the environment is predictable, the possible actions are limited and repeatable, the system is homogeneous in organisation, or the system is centralised. The domain of crisis management, however, violates all of these conditions. Crises are highly dynamic and chaotic, therefore solutions differ not only from one crisis to another, but a solution has to be continuously adapted to changes in the situation. Since in general multiple autonomous organisations are involved in dealing with a crisis, coordination of crisis response units is inherently distributed.

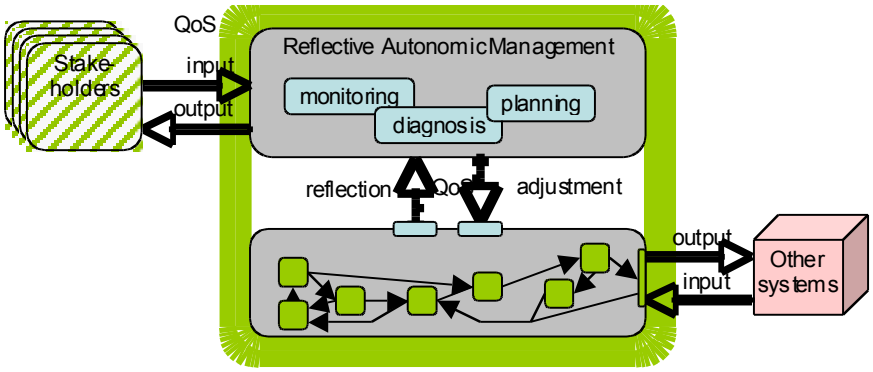


Figure 1: Representation of Reflective Autonomic Management component

This paper [1] proposes the Generic Reflective Autonomous Management (GRAM) architecture to support continuous adaptation of complex systems involving multiple organisations. The GRAM architecture respects differentiations in structure, goals and strategies within the different organisations. An autonomous entity, as shown in Figure 1, is modelled as a reflective self-managing unit and an organization as a layered composition of such units, as shown in Figure 2. At each layer a specific locally adequate management principle is be deployed. The exchange of policies and service level agreements between the layers is supported as well as the exchange of SLA and coordination information between layers of different organizations.

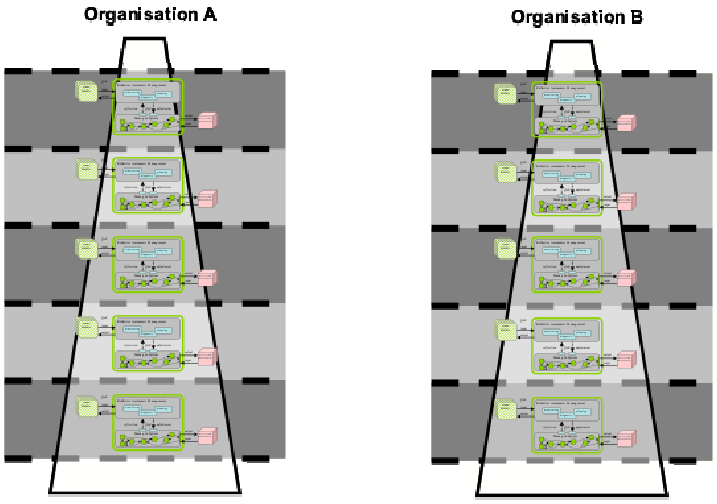


Figure 2: Representation of layered reflective organisations

A first approach to test and validate the GRAM Architecture is reported in this paper, bases on a prototype that integrates two different system management systems. The focus of the example scenario for this prototype revolves around coordination of fire-fighters in multiple safety regions. The GRAM prototype will be extended in cooperation with FP7 ALIVE project[1].

Acknowledgements

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (http://www.icis.decis.nl/), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The authors thank the NLnet Foundation (http://www.nlnet.nl) for supporting this work.

References

[1] J.B. van Veelen, S. van Splunter, N.J.E. Wijngaards, F.M.T. Brazier. Reconfiguration Management for Crisis Management Services. *The 15th conference of the International Emergency Management Society (TIEMS 2008)*, 2008.

[1] Project website: http://www.ist-alive.eu/

Decentralized Online Scheduling of Combination-Appointments in Hospitals[1]

Ivan Vermeulen[a] Sander Bohte[a] Sylvia Elkhuizen[b] Piet Bakker[b]
Han La Poutré[a]

[a] *Centrum Wiskunde & Informatica (CWI), Kruislaan 413, Amsterdam*
[b] *Academic Medical Center (AMC), Meibergdreef 9, Amsterdam*

1 Introduction

We research the online problem of scheduling combination appointments for outpatients at multiple departments. The service of scheduling a patient's appointments to a single day, a successful combination appointment, is high on the list of outpatient preferences and therefore of great importance to the hospital. It is hard to achieve for two reasons: first, due to the typical distributed authority in hospitals, scheduling combination appointments requires coordination between departments. Second, there is a trade-off between local scheduling efficiency and the fulfillment of patient scheduling preferences. Departments have local scheduling authority; each department is primarily concerned with its own performance. Making combination appointments requires coordination between departments, where each department solves a complex local scheduling problem.

Optimizing local performance is often complex. A department receives appointment-requests from multiple in- and outpatient departments with varying medical properties and urgencies. With a fixed resource capacity, appointments must be scheduled such that for all urgency levels a satisfactory fraction of patients is scheduled on time. The typical approach to this problem, is to classify patients into patient groups based on their urgency and medical properties, and allocate parts of the resource capacity to each patient group.

2 Approach

We present a multi-agent approach that respects the distributed nature of the hospital. Department agents represent local department scheduling objectives, and patient agents coordinate scheduling appointments by interacting with department agents. We consider the trade-off between local performance (measured by MSL: minimum service level) and making combination appointments (CA). Department agents rank all available timeslots with respect to schedule efficiency, and each department agent offers a set of the most efficient timeslots to the patient agent. The patient agent then selects and combines timeslots to make combination appointments. In our approach, timeslots are ranked based on a schedule-cost function.

We use a schedule-cost function for ranking timeslots: it assigns a cost value to each timeslot given the current patient to be scheduled and the current state of the calendar. In theory, if a cost function is optimally defined, selecting the timeslot with the lowest cost for each patient maximizes local performance (MSL). We identify some of the main properties of such a function, which generalizes and extends the ideas from [2], and design a parameterized function with those properties.

Let a scheduling-cost function $SC_d(p, ts)$ give the cost of scheduling patient p to timeslot ts in department d. The cost function $SC_d(p, ts)$ includes the following considerations: timeslots on days at the beginning of the schedule window have reduced cost to avoid wasting capacity; to balance the usage of timeslots over days, timeslots on days with many free timeslots have reduce cost; the cost of a timeslot is

[1] The full version of this paper is published as [1]

dependent on the patient group the timeslot is allocated to. These properties are combined in an overall schedule cost function.

We define a local scheduling method: First Come Maximal Relative Cost (**FCMRC(**mrc**)**). When scheduling a patient with **FCMRC(**mrc**)** the patient agent can select the preferred timeslot from all timeslots with a cost maximal the cost of the cheapest timeslots plus the parameter mrc (maximal relative cost). Using FCMRC(mrc), the trade-off between scheduling most efficiently and freedom in selecting a timeslot is set by parameter mrc.

We consider two methods for selecting a combination-appointment if there are multiple possibilities available.**COOR-LC:** the patient agent selects the combination-appointment with the lowest summed cost of individual timeslots, ensuring that the most efficient combination-appointment is selected. **COOR-RC:** the patient agent selects the preferred combination from all combination-appointments. This approach takes patient preferences into account but is less efficient than COOR-LC.

3 Experiments

All department agents have to schedule patient with varying urgencies, and use FCMRC(mrc) for local scheduling. We have manually determined the set of cost-function parameters that worked well for the whole range of settings of our presented experiments. We vary the parameter mrc to trade-off the opportunity for making combination appointment against local efficiency. We show the resulting trade-off by presenting Pareto fronts of $(CA, aMSL)$ solutions, in Figures 1a-b. We benchmark against approaches with an optimal static allocation of capacity: First Come First Serve (FCFS) without coordination and First Come Randomly Served (FCRS) with coordination.

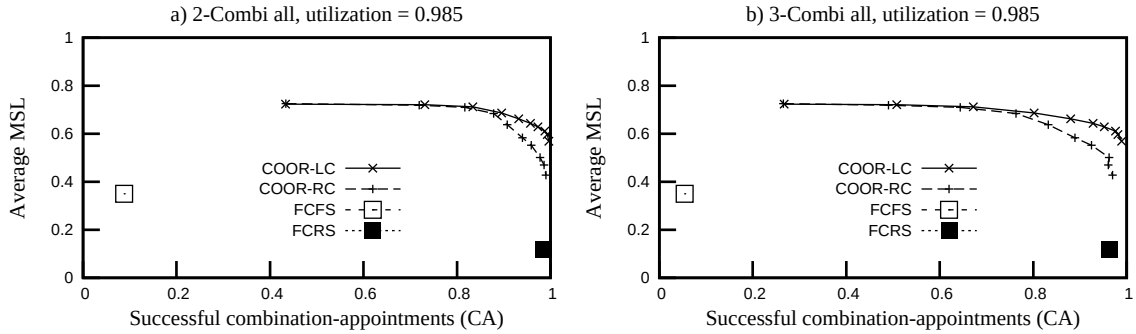


Figure 1: Trade-off between average local performance (MSL) and successful combination appointments (CA), for 2-combination appointments [a] and 3-combination appointments [b]

4 Conclusions

FCMRC(mrc) with varying mrc values, successfully finds efficient schedules, and allows a high performance trade-off between objectives. Our coordination approach allows scheduling combination appointments while maintaining desired performance levels at the local departments involved. Our approach allows a hospital or department to set a desired level of efficiency versus fulfilling patient preferences. In future work we additionally consider online mechanisms for dynamically setting this trade-off at each department.

References

[1] I.B. Vermeulen, S.M. Bohte, S.G. Elkhuizen, P.J.M. Bakker, and J.A. La Poutré. Decentralized online scheduling of combination-appointments in hospitals. In *Proceedings of the International Conference on Automated Planning and Scheduling*. AAAI Press, 2008.

[2] I.B. Vermeulen, S.M. Bohte, S.G. Elkhuizen, J.S. Lameris, P.J.M. Bakker, and J.A. La Poutré. Adaptive optimization of hospital resource calendars. In Proc. AIME, volume 4594 of *Lecture Notes in Computer Science*, pages 305–315. Springer, 2007.

Polynomial Distinguishability of Timed Automata[1]

Sicco Verwer [a] Mathijs de Weerdt [a] Cees Witteveen [a]

[a]*Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, the Netherlands*

1 Introduction

We are interested in identifying (learning) a system model for a real-time process. *Timed automata*(TAs) [1] are insightful models that can be used to model and reason about many real-time systems such as network protocols, business processes, reactive systems, etc. TAs are finite state models that model time *explicitly*, i.e. using numbers. In practice, it can be very difficult to construct a TA by hand. That is why we are interested in automatically identifying TAs from data. Such data can be obtained from sensors observing the process to be modeled. This results in a *time series of system states* or a *time-stamped event sequence*: every time-step the events occurring in the system are measured and recorded. From such timed data, we could have opted to identify an untimed model that models time *implicitly*, i.e. using states instead of numbers. Examples of such models are the *deterministic finite state automaton* (DFA) and the *hidden Markov model* (HMM). The reason for modeling time explicitly is that modeling time implicitly results in an exponential blow-up of the model size: numbers use a binary representation of time while states use a unary representation of time. Thus, an efficient algorithm that identifies a timed system using an untimed model is by definition an *inefficient* algorithm since it requires exponential time and space in the size of the timed model. Naturally, we would like our identification method to be efficient.

In this paper, we study the complexity of identifying (learning) TAs from data. We prove several theorems that set a bound on which types of TAs can, and which cannot, be identified efficiently from data. More specifically, we show that TAs can be identified efficiently only if they contain at most one timed component, known as a *clock*. Clocks are the time measuring objects in TAs. They can be thought of as stopwatches that can be reset by the state transitions of a TA. Boolean constraints on the values of these clocks are used to constrain the possible executions a TA. To the best of our knowledge, ours are the first results regarding the efficient identifiability (learnability) of TAs.

This paper is split into two parts. In the first part, we explain known results regarding the efficient identifiability of DFAs. We use these results to argue which types of TAs could be identified efficiently, and how to identify them. Then we describe our results regarding the efficient identifiability of TAs.

Efficient identification in the limit

We would like to have an efficient identification process for TAs. This is difficult due to the fact the identification problem for DFAs is NP-complete [4]. This property easily generalizes to the problem of identifying a TA (by setting all time values to 0). Thus, unless $P = NP$, a TA cannot be identified efficiently. Even more troublesome is the fact that the DFA identification problem cannot even be approximated within any polynomial [6]. Hence (since this also generalizes), the TA identification problem is also inapproximable.

These two facts make the prospects of finding an efficient identification process for TAs look very bleak. However, both of these results rely on there being a fixed input for the identification problem (encoding a hard problem). While in normal decision problems this is very natural, in an identification problem the amount of input data is somewhat arbitrary: more data can be sampled if necessary. Therefore, it makes

[1]The full paper is to appear in *Proceedings of the International Colloquium on Grammatical Inference, LNAI*, Springer, 2008.

sense to study the behavior of an identification process when is it given more and more data (no longer encoding the hard problem). The framework that studies this behavior is called *identification in the limit* [3].

This framework can be summarized as follows. Let C be a class of languages (for example the regular languages, modeled by DFAs). When given an increasing amount of examples from some language $L \in C$, a limit identification algorithm for C should at some point converge to L. If there exists such an algorithm A, C is said to be *identifiable in the limit*. If a polynomial amount of examples in the size of the smallest model for L is sufficient for convergence of A, C is said to be *identifiable in the limit from polynomial data*. If A requires time polynomial in the size of the examples, C is said to be *identifiable in the limit in polynomial time*. If both these statements hold, then C is *identifiable in the limit from polynomial time and data*, i.e. *efficiently identifiable in the limit*.

DFAs have been shown to be efficiently identifiable in the limit [5]. Also, it has been shown that *non-deterministic finite automata*(NFAs) are not efficiently identifiable in the limit [2]. This again generalizes to the problem of identifying a non-deterministic TA. Therefore, we only consider the identification problem for *deterministic timed automata* (DTAs).

Polynomial distinguishability of timed automata

Our goal is to determine exactly when and how DTAs can be identified efficiently in the limit. In this paper, we set a bound on which types of DTAs can, and which cannot, be identified efficiently in the limit. Our results are based on a property we call polynomial distinguishability. We call a class of automata C *polynomially distinguishable* if there exists a polynomial function p, such that for any two automata $\mathcal{A}, \mathcal{A}' \in C$ such that $L(\mathcal{A}) \neq L(\mathcal{A}')$, there exists a string $\tau \in L(\mathcal{A}) \triangle L(\mathcal{A}')$, such that $|\tau| \leq p(|\mathcal{A}| + |\mathcal{A}'|)$, where $L(\mathcal{A})$ is the language of $\mathcal{A}$. We use this property to show the following:

- Polynomial distinguishability is a necessary requirement for efficient identifiability in the limit.
- DTAs with at least two clocks are not polynomially distinguishable.
- DTAs with one clock are polynomially distinguishable.

These efficiency results have important consequences for anyone interested in identifying timed systems (and TAs in particular). Most importantly, they tell us that DTAs with one clock seem to be a good model for identifying a timed system from data. Furthermore, they show that anyone who needs to identify a DTA with two or more clocks should either be satisfied with sometimes requiring an exponential amount of data, or he or she has to find some other method to deal with this problem. This also holds for learning frameworks other then identification in the limit. The polynomial distinguishability result of 1-DTAs is necessary for the next step required to reach our goal, which is to write an algorithm that identifies 1-DTAs efficiently in the limit.

References

[1] Rajeev Alur and David L. Dill. A theory of timed automata.*Theoretical Computer Science*, 126:183–235, 1994.

[2] Colin de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27, 1997.

[3] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[4] E. Mark Gold.Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

[5] J. Oncina and P Garcia. Inferring regular languages in polynomial update time. In *Pattern Recognition and Image Analysis*, volume 1 of *Series in Machine Perception and Artificial Intelligence*, pages 49–61. World Scientific, 1992.

[6] Leonard Pitt and Manfred K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the ACM*, 40(1):95–142, 1993.

Decentralized Learning in Markov Games[1]

Peter Vrancx[a] pvrancx@vub.ac.be
Katja Verbeeck[b] katja.verbeeck@kahosl.be
Ann Nowé[a] ann.nowe@vub.ac.be

[a] *Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium*
[b] *Information Technology Group, KaHo Sint-Lieven, Ghent, Belgium*

Abstract

Reinforcement Learning (RL) was originally developed for Markov Decision Problems (MDPs) [9]. It allows a single agent to learn a policy that maximises a possibly delayed reward signal in a stochastic stationary environment. RL guarantees convergence to the optimal strategy as long as the agent can experiment enough and the environment in which it is operating has the Markov property.

However the MDP model does not allow multiple agents to act in the same environment. A straightforward extension of the MDP model to the multi-agent case is given by the framework of Markov Games [6]. In a Markov Game, actions are the result of the joint action selection of all agents and rewards and state transitions depend on these joint actions. When only one state is assumed, the Markov game is actually a repeated normal form game well known in game theory [8]. When only one agent is assumed, the Markov game is again an MDP.

Unfortunately, rewards are sensed for combinations of actions taken by different agents, and therefore agents are actually learning in a product or joint action space. Moreover, due to the existence of different reward functions, it generally is impossible to find policies which maximise the rewards for all agents. The latter is possible in the so-called team games or multi-agent MDPs (MMDPs). In this case, the Markov game is purely cooperative and all agents share the same reward function. In MMDPs the agents should learn to find and agree on the same optimal policy. In a general Markov game, an equilibrium point is sought; i.e. a situation in which no agent alone can change its policy to improve its reward when all other agents keep their policy fixed.

In addition, agents in a Markov game face the problem of incomplete information with respect to the action choice. One can assume that the agents get information about their own choice of action as well as that of the others. This is the case in what is called joint action learning, a popular way to address multi-agent learning [6, 2, 1, 3]. Joint action learners are able to maintain models of the strategy of others and explicitely take into account the effects of joint actions. In contrast, *independent agents* only know their own action. The latter is often a more realistic assumption since distributed multi-agent applications are typically subject to limitations such as partial or non observability, communication costs, asynchronism and stochasticity.

Different approaches toward independent multi-agent learning exist, however most of them have limited applicability or lack theoretical results. In [5] an algorithm is proposed for learning cooperative MMDPs, but it is only suited for deterministic environments. In [10, 4] new multi-agent exploration mechanisms are developed, however only single-state problems or repeated normal form games are considered.

In this paper we will focus on how learning automata (LA) can tackle the problem of learning Markov Games. Learning automata are independent, adaptive decision making devices that were previously shown to be very useful tools for building new multi-agent reinforcement learning algorithms in general [7]. The main reason for this is that even in multi-automata settings they still exhibit nice theoretical properties. One of the principal contributions of LA theory is that a set of decentralized learning automata is able to control a finite Markov Chain with unknown transition probabilities and rewards. In this paper we extend this result to the framework of Markov Games. A simple learning automaton is put for every agent in each state. This setting is analysed from 3 different perspectives; *the single superagent view*, in which a single agent is

[1] Paper published in IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 38 (4), August 2008

represented by the whole set of automata, *the multi-agent view*, in which each agent is represented by the automata it was associated with in each state and finally *the LA-view*, i.e. the view in which each automaton represents an agent. We show that under the same ergodicity assumptions of the original theorem, the multi-agent and LA view can both be approximated by a limiting normal form game, that share the same pure equilibrium points. Combined with previous results on automata games we will be able to prove that a set of independent learning automata agents is able to reach an equilibrium point in Markov Games.

Acknowledgements

Peter Vrancx funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen).

References

[1] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 709 – 716, Melbourne, Australia, 2003.

[2] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 746 – 752, 1998.

[3] J. Hu and M. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039 – 1069, 2003.

[4] S. Kapetanakis, D. Kudenko, and M. Strens. Learning to coordinate using commitment sequences in cooperative multi-agent systems. In *Proceedings of the 3rd Symposium on Adaptive Agents and Multi-agent Systems, (AISB03) Society for the study of Artificial Intelligence and Simulation of Behaviour*, 2003.

[5] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 535 – 542, 2000.

[6] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 322 – 328, 1994.

[7] A. Nowé, K. Verbeeck, and M. Peeters. Learning automata as a basis for multi-agent reinforcement learning. *Lecture Notes in Computer Science*, 3898:71–85, 2006.

[8] J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[9] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[10] Katja Verbeeck, Ann Nowe, J. Parent, and Karl Tuyls. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *The Journal of Autonomous Agents and Multi-Agent Systems*, 14-3:239–269, 2007.

Organized Anonymous Agents*

Martijn Warnier Frances Brazier

VU University, Amsterdam

Abstract

Anonymity can be of great importance in distributed agent applications such as e-commerce & auctions. This paper proposes and analyzes a new approach for organized anonymity of agents based on the use of pseudonyms. A novel naming scheme is presented that can be used by agent platforms to provide automatic anonymity for *all* agents on its platform, or, alternatively, to provide anonymity *on demand*. The paper also introduces a new technique, based on the use of handles, that can be fully integrated in an agent platform. Performance measures for an anonymity service implemented for the AgentScape platform provides some insight in the overhead involved.

This paper proposes a new approach to anonymous agent-to-agent communication that guarantees anonymity, (1) if required, for all agents running on a platform without any additional effort by agent application developers, or (2) on demand. The one main assumption is that agents trust the middleware on which they run –the agent platform. The link between an agent and its owner does not have to be anonymous: the middleware is trusted and can thus be trusted to keep this information confidential.

Anonymity in the real world is not an absolute notion, communication can be anonymous to one person or organisation, and not to another. Similarly, agents can communicate anonymously to other agents, or groups of agents, but not, for example, to the agent platform on which they run. Several degrees of anonymity can be distinguished ranging from absolute anonymity to total non-repudiation this paper focuses on *organized pseudonym-based semi-anonymity*. Semi-anonymity is organized: when and where anonymity is provided is well-defined. The naming scheme this paper introduces ensures that each agent's true identity and its pseudonyms are unique and cannot be linked to each other by any outside party. Classical anonymity in Computer Systems focuses on anonymity of the underlying communication layer, focusing on sender, receiver and link anonymity (unlinkability) respectively. The first two, sender and receiver anonymity, require that the location of the sender and receiver, respectively, are hidden from the other communicating party. Link anonymity, also known as unlinkability, ensures that the link between the communicating parties remains anonymous to all third parties: it is impossible for any outside party to observe if two parties are communicating with each other. (Note that the communicating parties themselves are often aware of each others (true) identity.) In practice these forms of anonymity can be combined.

This general notion of anonymity in Computer Science can be applied to agent technology. The focus in this paper is on anonymity of communication between individual agents. As stated earlier the relation between agent owner and agent is assumed to be confidential, and guaranteed by the agent platform. Full communication anonymity, i.e., receiver, sender and link anonymity taken together, can only be established when an agent cannot be linked to a legal entity via its communication with other agents. A platform based solution that enables the middleware to (automatically) provide link anonymity and location anonymity for each individual agent is the main focus of this paper. Pseudonyms are introduced for this purpose.

The use of a pseudonym, however, on its own does not suffice. If outsiders can observe agent communication, these observations can be used to obtain/deduct (unwanted) information about an agent. If an agent uses the same pseudonym to communicate with several other agents, together they can infer that they have been talking to the same party which breaks anonymity. An agent platform identifies an agent by its *globally unique identifier*(GUID). This GUID corresponds uniquely to the identity of the agent. The following example illustrates how the use of a single pseudonym for all communication does not suffice in multi-party negotiation situations:

*The full version [1] of this paper appeared in Proceedings of the Third International Symposium on Information Assurance and Security (IAS'07).

Example 1
There are three agents A, B and C, each with their own pseudonym P_A, P_B and P_C respectively. Agent A is interested in a service that both agent B or C can provide. Agent A first uses its pseudonym P_A to ask agent B about the price of its service, then agent A uses the same pseudonym P_A to ask agent C about the price of its services. Although agent B and agent C do not know A's real identity, together they can still determine that the same agent *has been asking price information of the services they provide. Thus agent A has not been communicating anonymously.*

Example 1 above clearly demonstrates the need for agents to use more than one pseudonym to obtain (link) anonymity – one pseudonym for each individual communication event (or communication session). For similar reasons, agents should also use a different pseudonym each time they communicate with the same party at some later point in time. The example also illustrates that privacy protection against buyer profiling cannot be obtained by solely using *one* pseudonym. As the same pseudonym can be linked to multiple events over a longer period of time, a buyer profile can be constructed, and privacy cannot be guaranteed. Even if the 'real' identity of the agent owner is not known!

In our approach each agent has one globally unique identifier and multiple unique pseudonyms (called *handles*). The agent platform is responsible for ensuring that all GUID's and pseudonyms are unique, that GUID's are hidden from other agents, that pseudonyms cannot be linked to each other and that pseudonyms cannot be linked to the agent they represent. This naming scheme is implemented using handles. Each agent is assumed to have a global unique identifier (GUID) known only to the agent platform. Such a GUID can, for example, be implemented by a Universally Unique Identifier (UUID, ISO 11578:1996). Furthermore, each agent can acquire as many (globally unique) handles as it requires. These handles serve as pseudonyms and are used for communication purposes.

As handles have no intrinsic meaning and do not leak any information about an agent or its owner, agents can safely use handles as pseudonyms. Link anonymity is acquired if agents use a new handle for each individual communication event. The agent platform is responsible for creation of agent GUID's and handles and the binding between the two. The binding between handles and GUID's can be acquired using a cryptographic hash function (sha): $\text{handle}_n = \text{sha}(\text{GUID} + \text{n})$ with $n \in \mathbb{N}^+$

This approach has two specific advantages: (i) if the GUID is not known then handles cannot be linked to each other or one specific GUID and (ii) if the GUID is known then the platform cannot deny that a specific handle belongs to a specific GUID.

An agent platform given an agent's handle, must be able to retrieve its GUID. A private lookup service provides this functionality. This service must be private to the middleware. Note, that an agent platform can always check the integrity of its own lookup service should it doubt the information it acquires. An agent platforms can always reconstruct an agent's handles given its GUID as described above and compare it to the handle it has been provided. As handles (or human readable names that are uniquely mapped to these handles) are used for all communication between agents, no information about the location of a particular agent is revealed to any other agent. Hence this technique also provides location anonymity and thus also sender and receiver anonymity. Whenever two agents communicate they do not have to share information on the location (host) on which they reside.

The proposed technique for acquiring anonymity, based on handles, can be completely integrated into agent platform middleware as a separate middleware service, (as described for AgentScape in [1]. The maximum performance penalty for this service is a factor of two overhead for agent communication.

References

[1] M. Warnier and F. M. T. Brazier. Organized anonymous agents. In *the Proceedings of The Third International Symposium on Information Assurance and Security (IAS'07)*. IEEE, 2007.

Topic Detection by Clustering Keywords

Christian Wartena [a] Rogier Brussee [a]

[a] *Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands*

Abstract

We consider topic detection without any prior knowledge of category structure or possible categories. Keywords are extracted and clustered based on different similarity measures using the induced k-bisecting clustering algorithm. Evaluation on Wikipedia articles shows that clusters of keywords correlate strongly with the Wikipedia categories of the articles. In addition, we find that a distance measure based on the Jensen-Shannon divergence of probability distributions outperforms the cosine similarity. In particular, a newly proposed term distribution taking co-occurrence of terms into account gives best results.

1 Introduction

In this paper we consider the problem of finding the set of most prominent topics in a collection of documents. Since we will not start with a given list of topics, we treat the problem of identifying and characterizing a topic as an integral part of the task. As a consequence, we cannot rely on a training set or other forms of external knowledge, but have to get by with the information contained in the collection itself. The approach we will follow consists of two steps. First we extract a list of the most informative keywords. Subsequently we try to identify clusters of keywords for which we will define a center, which we take as the representation of a topic. We found that this works fairly well in an evaluation with Wikipedia articles in which we compared human defined topic categories with keyword clusters.

Clustering of (key)words requires a similarity or distance measure between words. In this paper we consider distance measures between words that are based on the statistical distribution of words in a corpus of texts. The focus of is to find a measure that yields good clustering results.

2 Clustering keywords using co-occurrence distributions

We will represent a topic by a cluster of keywords. We therefore need a set of keywords for the corpus under consideration. The problem of finding a good set of keywords is similar to that of determining term weights for indexing documents, and not the main focus of this paper. For our purposes usable results can be obtained by selecting the most frequent nouns, verbs (without auxiliaries) and proper names and filtering out words with little discriminative power.

By clustering we will understand grouping terms, documents or other items together based on some criterion for similarity. We will define similarity using a distance function on terms, which we will in turn define as a distance between distributions associated to (key)words.

We compare results obtained using four different distance functions on keywords t and s: (a) the cosine similarity of the document distribution Q_t and Q_s considered as vectors on the document space, (b) the cosine similarity of the vectors of *tf.idf* values of keywords, (c) the Jensen-Shannon divergence between the document distributions of t and s and (d) the Jensen-Shannon divergence between the co-occrence distributions of t and s.

The co-occurrence distribution of a term is a probability distribution over terms defined in this paper that gives for each term a probability to co-occur with a given term. More precisely, the co-occurrence distribution is the weighted average of the term distributions of documents containing z where the weight is the probability $Q(d|z)$ that an instance of term z is contained in d.

For clustering we have used the induced bisecting k-means algorithm, which is based on the standard bisecting k-means algorithm.

3 Evaluation

To test and compare the different strategies we have compiled a corpus of Dutch Wikipedia articles consisting of 758 documents. In the analysis phase, 118099 term occurrences, and 26373 unique terms were found. The articles were taken from 8 Wikipedia categories: spaceflight, painting, architecture, trees, monocots, aviation, pop music, charadriiformes. Categories were selected for subjective similarity, like spaceflight and aviation, and subjective dissimilarity like pop music and monocots. Articles are equally distributed over the categories, but articles in some categories are significantly longer than in others. Moreover, homogeneity and specifity of articles differs significantly between categories.

To determine a set of relevant keywords we have selected the most frequent content words and filtered out a number of overly general terms. The latter was done by requiring that a keyword has to be different enough from the background distribution q, as measured by the Kullback-Leibler divergence.

Before extracting keywords we do some preprocessing like lemmatization, multiword lookup and named entity recognition. Tagging allows us to distinguish content words from function words. After lemmatization all inflected forms of verbs, nouns and adjectives are mapped onto their lexical form, substantially reducing the variation in the input data. While some of the components are language dependent, all of the components are available for a number of languages within the framework used for preprocessing.

To evaluate the implemented topic detection methods, we have compared the results with topics known to be present in the collection. We benchmarked against the 8 selected Wikipedia topics of the collection. Of course, it is conceivable that the collection has more topics that automatic methods might recognize as well. To define a reference clustering, we have clustered the 160 selected keywords into a set of 9 categories, one for each Wikipedia category and a rest cluster.

Since the clustering algorithm can produce clusters of different size, the quality of the result depends on the only input parameter of the algorithm, the threshold value to split up a cluster. We compare the different measures by their results for the samen number of clusters produced. For each number the measure based on the co-occurrence distribution gives best results.

4 Conclusion

The experimental results suggest that topic identification by clustering a set of keywords works fairly well, using either of the investigated similarity measures. In the present experiment a recently proposed distribution of terms associated with a keyword clearly gives best results. We believe that this is due to the fact that co-occurrence of a term with all other terms is taken into account.

References

[1] Christian Wartena and Rogier Brussee. Topic detection by clustering keywords. In *DEXA Workshops*, pages 54–58. IEEE Computer Society, 2008.

Modeling Agent Adaptation in Games*

Joost Westra Frank Dignum Virginia Dignum

Utrecht University
P.O. Box 80.089, 3508 TB, The Netherlands

1 Introduction

Increasing complexity of software games, in particular of serious games [3], and the need to reuse and adapt games to different purposes and different user needs, requires distributed development approaches. As games become more and more sophisticated in terms of graphical and technological capabilities, higher demands are also put on the content provided by them. With content we mean the story, characters, dialogue, game play, music, sound etc. Once the complexity of the games increases and more natural behavior of characters and other elements of the game is required the use of agents becomes a good option. This holds especially for serious games which usually are very complex with different characters performing complicated tasks that need to adjust to the trainee. E.g. a character being more smart or getting new capabilities to keep the game challenging. It would be very useful if this could be done without rewriting the complete strategy.

In this paper we propose a system for adapting agents in an organized fashion. The reason is that we need the agents to adapt to the user while he is playing the computer game (during gameplay) and we want to make sure the game does not reach unwanted states. This is less of a problem if the learning is done offline, then you just want to make sure that the end result is good. In current computer games (adaptive) agents are hardly used because control of the game is a very important aspect. Designers of computer games put a lot of effort in creating scenario's and creating a certain storyline, which should not be disturbed by randomly adapting agents.

We propose to view games as designed organizations developed to realize a specific set of goals and requirements (typically the learning objectives for serious games). This explicit organizational structure is then elaborated by the individual agents into appropriate behaviors. Global game structure is determined by organizational design, and is independent of the agents themselves. Such structures implement the idea that agent interactions occur not just by accident but aim at achieving some desired global goals. That is, there are goals external to each individual agent that must be reached by the interaction of those agents

The Agent Organization [1] is used to describe game objectives in terms of the capabilities and constraints of organizational concepts such as roles (or function, or position), groups (or communities), tasks (or activities) and interaction protocols (or dialogue structure), thus on what relates the structure of an organization to the externally observable behavior of its agents. Agents are autonomous, so can in principle evaluate organization state and decide to adapt their behavior, but only within the organizational constraints.

2 Agent-Based Game Adaptation

A lot of research has already been done on adapting separate agents. These techniques are very relevant to our research because even though the agents are guided by the organization they still adapt individually. Using common reinforcement-learning algorithms [4] agents are able to do very complex adaptations. A lot of the reinforcement-learning techniques are more suitable for offline adaptation but they can also be used for online adaptation. Another possibility to do the adaptation is to use genetic algorithms. Most

*The full version of this paper appeared in the OAMAS workshop at AAMAS, May 2008, Lisbon. This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

research on genetic algorithms is also done on offline learning. But some online learning algorithms have been developed. Some are even already applied in the area of computer games.

Current approaches using multi-agent adaptation use either individual adaptation without any direct coordination. Or they use coordination that requires all the agents to know a lot of information about the other agents. This approach can only be used in very simple environments. Most games are very complex environments making the approach, where all the agents have to coordinate with each other, very impractical. In our approach we let all the agents adapt individually but guiding the adaptation using an agent organization. We use OperA [1] as the framework for the design of adaptable games.

The OperA model for agent organizations enables the specification of organizational requirements and objectives, and at the same time allows participants to have the freedom to act according to their own capabilities and demands. The OperA approach separates the concerns of the organization from those of the individual. The top layer, the Organizational Model (OM), describes the structure and objectives of a system as envisioned by the organization, in terms of roles, norms and interactions.

Individuals have however their own expectations and requirements that do not always fit organizational design. The Social Model (SM), describes the agent population in terms of the agreements made between individual agents and the overall organization. Social contracts describe the capabilities and responsibilities of an agent within the society, that is the desired way that an agent will fulfill its role(s).

Using OperA enables us to describe the storyline of the game by means of global role objectives and behavioral norms. In serious games it also allows to divide tasks over landmarks that can be arranged in a partial ordered graph, thus directing the overall learning experience.

We are using OperA to describe the organizational structures of the game, but the actual execution of all the tasks is done by the agents enacting the roles. As a possible implementation for the agents that will act in the coordination model we are investigating the usage of adaptable BDI agents. For this purpose we are using 2APL; an agent-oriented programming language. It provides the functionality to program cognitive agents based on a BDI architecture. This makes it possible to design agents that are able to reason and to plan. Normally when executing a 2APL agent the first possible applicable plan in the plan-base is always selected. In computer games there often are a lot of different possible plans or actions that can be chosen at the same time. Some of these actions will be better than other actions. We have created an extension [2] to the 2APL language to make it possible to learn a preference relation between the different possible rules given a certain situation. We have used an approach similar to dynamic scripting. As we described earlier this will also make it possible for the agents to adapt to the user. This preference on the different rules is not done by only adjusting the weights but is also dependent on the current environmental information. There are different techniques that can be used to do this adaptation. We investigated a Monte Carlo based method and temporal difference learning (sarsa on-policy) but it is beyond the scope of this article to go in the details of this experiment.

3 Conclusion

In this abstract we have shown that it would be useful to create a system for creating training games that dynamically adapt to the human user. We use autonomous agents for all the possible adaptable elements and show how the OperA agent coordination model can be used to organize and control the autonomous agents. Using OperA makes it possible to define the overall requirements and flow of the total game, and still make it possible for the different elements to learn and adapt within these limitations.

References

[1] V. Dignum. A Model for Organizational Interaction: based on Agents, founded in Logic. *SIKS Dissertation, series*.

[2] Eric Kok. Learning Effective Rule Selection in 2APL Agents . 2007.

[3] B.G. Silverman, G. Bharathy, K. O'Brien, and J. Cornwell. Human behavior models for agents in simulators and games: part II: gamebot engineering with PMFserv. *Presence: Teleoperators and Virtual Environments*, 15(2):163–185, 2006.

[4] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Monte-Carlo Tree Search Solver[1]

Mark H.M. Winands [a] Yngvi Björnsson [b] Jahn-Takeshi Saito [a]

[a] *Games and AI Group, MICC, Universiteit Maastricht, Maastricht, The Netherlands*
{m.winands,j.saito}@micc.unimaas.nl
[b] *School of Computer Science, Reykjavík University, Reykjavík, Iceland*
yngvi@ru.is

For decades $\alpha\beta$ search has been the standard approach used by programs for playing two-person zero-sum games such as chess and checkers. Over the years many search enhancements have been proposed for this framework. However, in some games where it is difficult to construct an accurate positional evaluation function (e.g., Go) the $\alpha\beta$ approach was hardly successful. In the past, Monte-Carlo (MC) methods have been used as an evaluation function in a search-tree context [1, 3, 4]. A direct descendent of that approach is a new general search method, called Monte-Carlo Tree Search (MCTS) [6, 9]. MCTS is not a classical tree search followed by a MC evaluation, but rather a best-first search guided by the results of Monte-Carlo simulations. It is based on a randomized exploration of the search space. Using the results of previous explorations, the algorithm gradually builds up a game tree in memory, and successively becomes better at accurately estimating the values of the most promising moves. In the last two years MCTS has advanced the field of computer Go substantially. Moreover, it is used in other games as well (Phantom Go [5], Clobber [10]), even for games where there exists already a reasonable evaluation function (e.g., Amazons [8]).

MCTS consists of four strategic steps, repeated as long as there is time left. The steps are as follows. (1) In the *selection step* the tree is traversed from the root node until we reach a node *E*, where we select a position that is not added to the tree yet. (2) Next, during the *play-out step* moves are played in self-play until the end of the game is reached. The result R of this "simulated" game is $+1$ in case of a win, 0 in case of a draw, and -1 in case of a loss. (3) Subsequently, in the *expansion step* children of E are added to the tree. (4) Finally, R is propagated back along the path from *E* to the root node in the *backpropagation step*. When time is up, the move played by the program is the child of the root with the highest value.

MCTS is unable to *prove* the game-theoretic value of (even parts of) the search tree, but in the long run MCTS equipped with the UCT formula [9] is able to *converge* to the game-theoretic value. For a fixed termination game like Go, MCTS is able to find the optimal move relatively fast [15]. But in a sudden-death game like chess or Lines of Action (LOA), where the main line towards the winning position is narrow, MCTS may often lead to an erroneous outcome because the nodes' values in the tree do not converge fast enough to their game-theoretical value. We designed a new variant called MCTS-Solver, which is able to prove the game-theoretical value of a position. This is an important step towards applying MCTS-based approaches effectively to sudden-death like games (including chess). The standard backpropagation and selection mechanisms have been modified for this variant.

In this paper we use the game LOA as a testbed. It is an ideal candidate because its intricacies are less complicated than those of chess. So, we can focus on the sudden-death property. Furthermore, because LOA was used as a domain for various other AI techniques [2, 7, 12], the level of the state-of-the-art LOA programs is high, allowing us to look at how MCTS approaches perform against increasingly stronger evaluation functions. Moreover, the search engine of a LOA program is quite similar to the one of a chess program. Our experiments show that a MC-LOA program using MCTS-Solver defeats the original MCTS program by an impressive winning score of 65%. Furthermore, when playing against a state-of-the-art $\alpha\beta$-based program MIA [12] equipped with different evaluation functions [11, 13, 14], MCTS-Solver performs better than a regular MCTS program (see Table 1).

Thus, we may conclude that MCTS-Solver is a genuine improvement, significantly enhancing MCTS in sudden-death games. Although MCTS-Solver is still trailing behind the best $\alpha\beta$-based program, we view

[1] The full version of this paper is published in: *Computers and Games (CG 2008)*, LNCS 5131, pp. 25–36, Springer, Heidelberg, Germany.

Table 1: 1,000-game match results against MIA with different evaluation functions

Evaluator	MIA 2000	MIA 2002	MIA 2006
MCTS	585.5	394.0	69.5
MCTS-Solver	692.0	543.5	115.5

this work as one step towards the goal of successfully applying simulation-based approaches in a wider variety of games. For these methods, to be able to handle proven outcomes is an essential step to make. With continuing improvements it is not unlikely that in the not so distant future enhanced simulation-based approaches may become a competitive alternative to $\alpha\beta$ search in games dominated by the latter so far.

References

[1] B. Abramson. Expected-outcome: A general model of static evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):182–193, 1990.

[2] D. Billings and Y. Björnsson. Search and knowledge in Lines of Action. In H.J. van den Herik, H. Iida, and E.A. Heinz, editors, *Advances in Computer Games 10: Many Games, Many Challenges*, pages 231–248. Kluwer Academic Publishers, Boston, MA, USA, 2003.

[3] B. Bouzy and B. Helmstetter. Monte-Carlo Go Developments. In H.J. van den Herik, H. Iida, and E.A. Heinz, editors, *Advances in Computer Games 10: Many Games, Many Challenges*, pages 159–174. Kluwer Academic Publishers, Boston, MA, USA, 2003.

[4] B. Brügmann. Monte Carlo Go. Technical report, Physics Department, Syracuse University, 1993.

[5] T. Cazenave and J. Borsboom. Golois Wins Phantom Go Tournament. *ICGA Journal*, 30(3):165–166, 2007.

[6] 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, *Proceedings of the 5th International Conference on Computer and Games*, volume 4630 of *Lecture Notes in Computer Science (LNCS)*, pages 72–83. Springer-Verlag, Heidelberg, Germany, 2007.

[7] B. Helmstetter and T. Cazenave. Architecture d'un programme de Lines of Action. In T. Cazenave, editor, *Intelligence artificielle et jeux*, pages 117–126. Hermes Science, 2006. In French.

[8] J. Kloetzer, H. Iida, and B. Bouzy. The Monte-Carlo Approach in Amazons. In H.J. van den Herik, J.W.H.M. Uiterwijk, M.H.M. Winands, and M.P.D. Schadd, editors, *Proceedings of the Computer Games Workshop 2007 (CGW 2007)*, pages 185–192, Universiteit Maastricht, Maastricht, The Netherlands, 2007.

[9] 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.

[10] L. Kocsis, C. Szepesvári, and J. Willemson. Improved Monte-Carlo Search, 2006. http://zaphod.aml.sztaki.hu/papers/cg06-ext.pdf.

[11] M.H.M. Winands. Analysis and implementation of Lines of Action. Master's thesis, Universiteit Maastricht, Maastricht, The Netherlands, 2000.

[12] M.H.M. Winands. *Informed Search in Complex Games*. PhD thesis, Universiteit Maastricht, Maastricht, The Netherlands, 2004.

[13] M.H.M. Winands, L. Kocsis, J.W.H.M. Uiterwijk, and H.J. van den Herik. Temporal difference learning and the Neural MoveMap heuristic in the game of Lines of Action. In Q. Mehdi, N. Gough, and M. Cavazza, editors, *GAME-ON 2002*, pages 99–103, Ghent, Belgium, 2002. SCS Europe Bvba.

[14] M.H.M. Winands and H.J. van den Herik. MIA: a world champion LOA program. In *The 11th Game Programming Workshop in Japan (GPW 2006)*, pages 84–91, 2006.

[15] P. Zhang and K. Chen. Monte-Carlo Go tactic search. In P. Wang et al., editors, *Proceedings of the 10th Joint Conference on Information Sciences (JCIS 2007)*, pages 662–670. World Scientific Publishing Co. Pte. Ltd., 2007.

Demonstrations

BNAIC 2008

Automatic Generation of Japanese Puzzles

K.J. Batenburg [a] W.A. Kosters [b]

[a] *Vision Lab, University of Antwerp, Belgium;*
joost.batenburg@ua.ac.be
[b] *Leiden Institute of Advanced Computer Science (LIACS),*
Universiteit Leiden, The Netherlands;
kosters@liacs.nl

1 Introduction

Japanese puzzles, also known as Nonograms, are logic puzzles that are sold by many news paper vendors. The challenge is to fill a grid with black and white pixels in such a way that a given description for each row and column, indicating the lengths of consecutive segments of black pixels, is adhered to. The resulting black-and-white image is typically a picture of an object, person or scenery. Although the general problem of solving Japanese puzzles is NP-hard [1, 4], the Nonograms in puzzle books can usually be solved by hand. With a few exceptions, they belong to a specific class of nonograms that can be solved by considering the information in a single row or column at a time.

2 Creating nonograms

Creating interesting Japanese puzzles is a cumbersome task [3]. The black-and-white image has a low resolution, yet the depicted scene should still be interpretable once the puzzle has been solved. Simple thresholding of greylevel images usually does not yield puzzles that can be solved by hand. Either the resulting puzzles do not have a unique solution, or branching is required to find the solution.

We have developed an algorithm for automatic generation of Japanese puzzles, based on a grey level input image. From a single input image, the algorithm can compute a list of puzzles of various difficulty levels that resemble this image. An important subroutine in our algorithm is the *puzzle solver* [2], which can quicky determine if a puzzle can be solved without branching, and how difficult the puzzle is. The very short running time of our solver (typically less than 1ms) allows the puzzle creation algorithm to effectively search the space of candidate puzzles that resemble the input image.

3 Demonstration

We have implemented a program to generate Japanese puzzles from arbitrary input images. The resulting Japanese puzzle can be stored as a pdf-file, presented in a format similar to the Japanese puzzle books sold in stores. At the conference, we will equip our system with a digital camera, so that participants can generate Japanese puzzles of themselves, or of arbitrary pictures they can provide on a USB-stick. The puzzles can also be printed, for those who cannot wait to start solving their puzzle.

(a) Original grey level image (b) Generated puzzle

Figure 1: The Japanese puzzle is generated based on a grey level input image and a set of algorithm parameters.

References

[1] K. J. Batenburg, W. A. Kosters, A discrete tomography approach to Japanese puzzles, in: Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC) (2004) 243–250.

[2] K. J. Batenburg, W. A. Kosters, A reasoning framework for solving Nonograms, Lect. Notes Comp. Sci. 4958 (2008) 372–383.

[3] E. Ortiz-Garcia, S. Salcedo-Sanz, J. Leiva-Murillo, A. Perez-Bellido, J. Portilla-Figueras, Automated generation and visualization of picture-logic puzzles, Computers and Graph. 31 (2007) 750–760.

[4] N. Ueda, T. Nagao, NP-completeness results for Nonogram via parsimonious reductions, preprint. URL: `citeseer.ist.psu.edu/ueda96npcompleteness.html` (1996). `citeseer.ist.psu.edu/ueda96npcompleteness.html`

Monte-Carlo Tree Search: A New Framework for Game AI[1]

Guillaume Chaslot, Sander Bakkes, Istvan Szita and Pieter Spronck

Universiteit Maastricht / MICC
P.O. Box 616, NL-6200 MD Maastricht, The Netherlands
{g.chaslot,s.bakkes,i.szita,p.spronck}@micc.unimaas.nl

Abstract

In this paper, we put forward Monte-Carlo Tree Search as a novel, unified framework to game AI, which doesn't require an evaluation function. In the framework, randomized explorations of the search space are used to predict the most promising game actions. We will demonstrate that Monte-Carlo Tree Search can be applied effectively to (1) classic board-games, (2) modern board-games, and (3) video games.

1 Introduction

When implementing AI for computer games, the most important factor is the evaluation function that estimates the quality of a game state. However, building an *adequate* evaluation function based on heuristic knowledge for a non-terminal game state is a domain-dependent and complex task. It probably is one of the main reasons why game AI in complex game-environments did not achieve a strong level, despite intensive research and additional use of knowledge-based methods. Monte-Carlo Tree Search (MCTS), a Monte-Carlo based technique that was first established in 2006, is implemented in top-rated GO programs. These programs defeated for the first time professional GO players on the 9×9 board. The technique can be generalized easily to modern board-games or video games. In the proposed demonstration, we will illustrate that MCTS can be applied effectively to (1) classic board-games (such as GO), (2) modern board-games (such as SETTLERS OF CATAN), and (3) video games (such as the SPRING RTS game).

2 Monte-Carlo Tree Search

Monte-Carlo Tree Search (MCTS) is a best-first search technique which uses stochastic simulations. MCTS can be applied to any game of finite length. Its basis is the simulation of games where both the AI-controlled player and its opponents play random moves, or, better, pseudo-random moves. From a single random game (where every player selects his actions randomly), very little can be learnt. But from simulating a multitude of random games, a good strategy can be inferred. The algorithm builds and uses a tree of possible future game states, according to a four-step mechanism. First, *selection*: while the state is found in the tree, the next action is chosen according to the statistics stored, in a way that balances between exploitation and exploration. On the one hand, the task is often to select the game action that leads to the best results so far (exploitation). On the other hand, less promising actions still have to be explored, due to the uncertainty of the evaluation (exploration). Several effective strategies can be found in Chaslot et al. [2] and Kocsis and Szepesvári [3]. Second, *expansion*: when the game reaches the first state that cannot be found in the tree, the state is added as a new node. This way, the tree is expanded by one node for each simulated game. Third, *simulation*: for the rest of the game, actions are selected at random until the end of the game. Naturally, the adequate weighting of action selection probabilities has a significant effect on the level of play. If all legal actions are selected with equal probability, then the strategy played is often weak, and the level of the Monte-Carlo program is suboptimal. We can use heuristic knowledge to give larger weights to actions that

[1]This research was funded by NWO grant No 612.066.406 and grant No 612.006.409.

look more promising. Fourth, *backpropagation*: after reaching the end of the simulated game, we update each tree node that was traversed during that game. The visit counts are increased and the win/loss ratio is modified according to the outcome.

3 Applications

Classic Board-Games such as two-player deterministic games with perfect information, have been submitted to intensive AI researched. Using the alpha-beta framework, excellent results have been achieved in the game of CHESS and CHECKERS. However, alpha-beta only works well under two conditions: (1) an adequate evaluation function exists, and (2) the game has a low branching factor. These two conditions are lacking in numerous classical board-games (such as GO), modern board-games and video games. As an alternative to alpha-beta, researchers opted the use of MCTS. It has been shown that MCTS is able to use highly randomised and weakly simulated games in order to build the most powerful GO-programs to date. In our demonstration, we will present our program MANGO, which is a top-rated GO program. We will use graphical tools to demonstrate how MANGO focuses its search on the best moves. We will emphasise that MCTS without any expert knowledge can still achieve a reasonable level of play.

Modern board-games are becoming more and more popular since their (re)birth in the 1990's. The game SETTLERS OF CATAN can be considered an archetypical member of the genre. Modern board-games are of particular interest to AI researchers because they provide a direct link between classic (two-player, perfect information) board-games and video games. On the one hand, state variables of most modern board-games are discrete, and decision making is turn-based. On the other hand, the gameplay in modern board-games often incorporates randomness, hidden information, multiple players, and a variable initial setup that makes it impossible to use opening books. In our demonstration, we will show that MCTS outperforms previous heuristic game AI's in SETTLERS OF CATAN, and provides a challenging opponent for humans.

Video games present a complex and realistic environment in which game AI is expected to behave realistically. When implementing AI in video games, arguably the most important factor is the evaluation function that rates the quality of newly generated game AI. Due to the complex nature of video games, the determination of an adequate evaluation function is often a difficult task. Still, experiments performed in the SPRING RTS game have shown that is is possible to generate an evaluation function that rates the quality of game AI accurately before half of the game is played [1]. However, it is desirable that accurate ratings are established even more early, when adaptations to game AI can influence the outcome of a game more effectively. Monte-Carlo simulations provide a powerful means to accurately rate the quality of newly generated game AI, even early in the game. In our demonstration, we will show how we abstract the SPRING RTS game for use of MCTS simulation. The abstraction contains, among others, the position of each unit in the game, and the game strategy employed by all players. We will emphasis that in complex video-games, effective game AI may be established by using MCTS, even with highly randomised and weakly simulated games.

4 Conclusions

In this abstract, we put forward Monte-Carlo Tree Search (MCTS) as a novel, unified framework to game AI. In the framework, randomized explorations of the search space are used to predict the most promising game actions. We state that MCTS is able to use highly randomised and simulated games in order to established effective game AI. In demonstrations, we will show that MCTS can be applied effectively to (1) classic board-games, (2) modern board-games, and (3) video games.

References

[1] Sander Bakkes and Pieter Spronck. *AI Game Programming Wisdom 4*, chapter Automatically Generating Score Functions for Strategy Games, pages 647–658. Charles River Media, Hingham, MA., U.S.A., 2008.

[2] G.M.J-B. Chaslot, J-T. Saito, B. Bouzy, J.W.H.M. Uiterwijk, and H.J. van den Herik. Monte-Carlo Strategies for Computer Go. In *Proceedings of the 18th Belgian-Dutch Conference on Artificial Intelligence*, pages 83–90, 2006.

[3] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006, Lecture Notes in Artificial Intelligence 4212*, pages 282–293, 2006.

Multimodal Interaction with a Virtual Guide

Dennis Hofs Mariët Theune Rieks op den Akker

University of Twente, P.O.Box 217, 7500 AE Enschede

Abstract

We demonstrate the Virtual Guide, an embodied conversational agent that gives directions in a 3D environment. We briefly describe multimodal dialogue management, language and gesture generation, and a special feature of the Virtual Guide: the ability to align her linguistic style to the user's level of politeness.

1 Introduction

At the University of Twente we have developed the Virtual Guide, an embodied conversational agent that can give route directions in a 3D virtual building called the Virtual Music Centre (VMC).[1] When navigating through the VMC, the user can approach the Virtual Guide to ask for directions. Currently the Virtual Guide is located at the reception desk of the VMC (see Figure 1), but she could be situated anywhere in the building. In fact, with only minor changes she could also be used for direction giving in actual environments.

2 The Virtual Guide

The first part of the interaction between the Virtual Guide and the user consists of a natural language dialogue in which the multimodal dialogue management module tries to find out the user's intended destination. This may involve subdialogues, in which either the Guide or the user asks the other for clarification, and the resolution of anaphoric expressions (e.g., *How do I get there?*).[2] Available input modalities are typed text or speech in combination with mouse pointing. To process the user's input, the Virtual Guide incorporates a speech recognizer (Philips SpeechPearl), a parser making use of a Dutch unification grammar, and a fusion module that merges deictic expressions with any co-occurring pointing gestures (e.g., the user asking *What is this?* while pointing at the VMC map). The results of input analysis are sent to a dialogue act classifier, which maps the user's utterance to one or more dialogue acts. Based on this, the dialogue manager chooses an appropriate action to be performed by the Virtual Guide, such as uttering a certain dialogue act (realised in natural language using one of a collection of sentence templates) or showing something on the map.

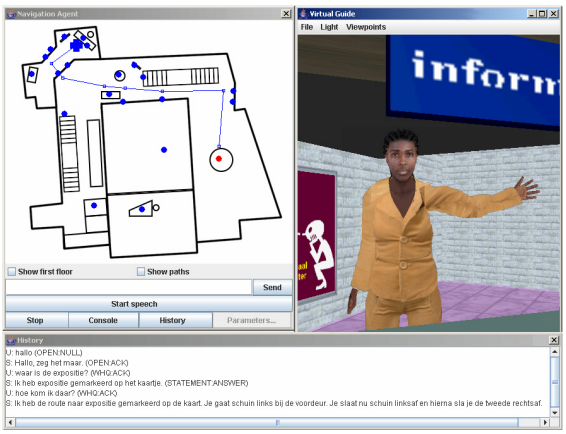


Figure 1: The Virtual Guide.

Recently, the Virtual Guide has been extended with an alignment module that enables dynamic adaptation of the Virtual Guide's linguistic style to that of the user. The grammar rules used for user input analysis have been associated with tags indicating the level of politeness of the user utterance, depending on the grammatical construction used. For example, an imperative such as *Show me the hall* is considered quite impolite, while indirect requests such as *I would like to know where the hall is* are considered very polite.

[1] The Virtual Guide is accessible online via http://wwwhome.cs.utwente.nl/~hofs/dialogue.

[2] The actual language of the Virtual Guide is Dutch, but for ease of reading examples in this paper are given in English.

The templates used to generate system utterances have been similarly tagged, allowing the Virtual Guide to adapt the politeness of its replies to that of the user. Using different parameter settings for the system's initial levels of politeness, as well as the degree of alignment, allows us to model different professional attitudes or personalities for the Guide.

Currently, the alignment module is only used for the dialogue part of the interaction, not for the actual generation of the route description, which is presented in the form of a monologue when the user's destination has been established. The route description consists of a sequence of segments consisting of a turn direction combined with a description of a landmark where this turn is to be made. For example, *You go left at the information sign*. For the generation of the route description, a template-based realisation component has been built based on Exemplars [4].

Finally, the Virtual Guide's gesture generation component extends the generated text with tags associating the words in the route description with appropriate gestures. The marked-up text is sent to the animation planner (based on [3]), which realises the required animations in synchronization with the Guide's speech output. For text-to-speech synthesis, either Loquendo[3] or Fluency[4] can be used. The 3D model used for the body of the Virtual Guide was purchased from aXYZ design.[5] In addition to being presented in speech and gesture by the Virtual Guide, the recommended route is also displayed on a 2D map of the VMC.

For more details on dialogue management, language generation and gesture generation in the Virtual Guide, see [2]. The linguistic alignment module used in the Virtual Guide is described in [1].

3 The demonstration

In the demonstration, which will last 10 to 20 minutes, we will carry out some scripted example interactions with the Virtual Guide to illustrate dialogue features such as multimodal fusion, resolution of anaphors and elliptic utterances, clarification subdialogues, error recovery and politeness alignment. In addition, visitors will be given the opportunity to interact freely with the Virtual Guide.

The system runs on a Windows computer with 2 GB of memory and a broadband Internet connection. It uses Java, Java 3D and Java Advanced Imaging.

Acknowledgements

We thank Martin Bouman and Richard Korthuis for their work on the language generation component, Marco van Kessel for his work on the gestures and embodiment, and Markus de Jong for developing the linguistic alignment component of the Virtual Guide. This work was carried out within the NWO project ANGELICA (grant no. 632.001.301).

References

[1] Markus De Jong, Mariët Theune, and Dennis Hofs. Politeness and alignment in dialogues with a virtual guide. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, volume 1, pages 207–214, 2008.

[2] Mariët Theune, Dennis Hofs, and Marco van Kessel. The Virtual Guide: A direction giving embodied conversational agent. In *Proceedings of Interspeech 2007*, pages 2197–2200, 2007.

[3] Herwin van Welbergen, Anton Nijholt, Dennis Reidsma, and Job Zwiers. Presenting in virtual worlds: Towards an architecture for a 3D presenter explaining 2D-presented information. *IEEE Intelligent Systems*, 21(5):47–53, 2006.

[4] Michael White and Ted Caldwell. EXEMPLARS: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-98)*, pages 266–275, 1998.

[3] http://www.loquendo.com/
[4] http://www.fluency.nl/
[5] http://www.axyz-design.com/

DEIRA: A Dynamic Engaging Intelligent Reporter Agent (Demo Paper)

François L.A. Knoppel Almer S. Tigelaar Danny Oude Bos Thijs Alofs

Human Media Interaction, University of Twente, P.O.Box 217 7500AE Enschede

Abstract

DEIRA is an embodied agent with a highly modular design, supporting several domains such as real time virtual horse race commentary, robosoccer commentary and virtual storytelling. Domain-specific information is processed to have the agent act on emotion and produce a compelling report on the situation, using synthesized speech and facial expressions. This paper briefly describes the features of the agent.

1 Introduction

DEIRA was originally developed as an embodied agent that provides commentary for real-time virtual horse races. Its first public appearance was at the IVA07 conference where it participated in the GALA07 contest, having to report on a race script supplied on the spot. The successful result, being a jury award shared with another contestant as well as the public award, combined with the modular design inspired further development into a platform for transforming domain-specific information into a compelling live report by an embodied agent.

2 System description

The following is a brief description of the system. A more elaborate version is available in [1].

The first step in the process is the transformation of the domain-specific information into meaningful events. Different domains ask for a different method of determining what is meaningful and consequently, for every domain, a domain-specific Input Analysis Module (IAM) exists. These modules are required to deliver the events in a specific format, enabling further processing.

For all the events, the Mental Model Module (MMM) determines the emotional impact of each event based on the event information and personality parameters of the reporter. It also maintains a general emotional state which is the combined result of all events that have occurred taking into account a reduced influence over time and the assigned importance of the events.

After passing the MMM the events are stored in a prioritized Event Queue (EQ), with the prioritization based on the importance and decay factor parameters the IAM has linked to the events. Updating the importance using the decay factor ensures that the events reported on first are indeed the

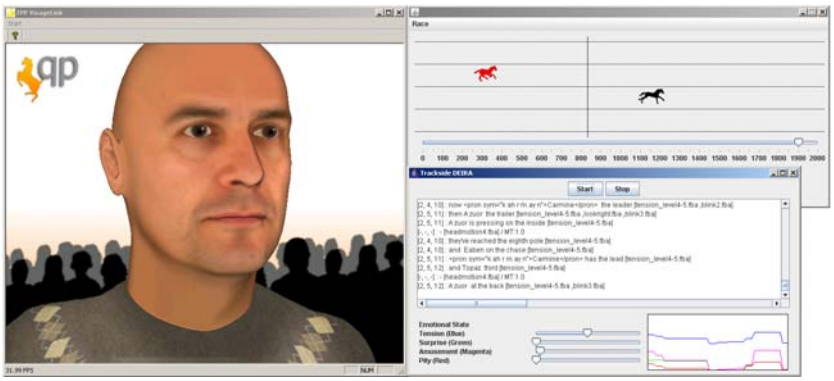


Figure 1: Screenshot of the system

most important ones. The EQ also decreases chances of repetition by lowering the importance of events that are similar to those recently uttered.

Using a generative context-free grammar supporting variables and conditionals, the Text Generation Module (TGM) constructs a set of potential utterances for each event it retrieves from the EQ. To prevent repetition in verbal content, a history of utterances is maintained. The grammar provides a rich vocabulary to report on each event, and is easy to expand or adjust for other purposes.

For vocal expression, the Speech Adaptation Module (SAM) subsequently determines at what speed, pitch and volume the sentences should be uttered based on the emotional state of the reporter and the emotional content of the event itself.

The facial expression of the agent is handled by the Facial Animation Module (FAM) in two ways. Secondary head animations (like saccadic eye movements and small head motions) are triggered at fixed intervals. When reporting on a specific event the emotional state of the reporter is used to generate an appropriate primary animation like smiling and frowning.

When all aspects of the output have been determined, an Output Module (OM) feeds the text plus utterance characteristics and animation data to an external application responsible for actually displaying the animated model and cooperating with a Text-To-Speech (TTS) engine to generate lip-synced speech.

The external applications we currently support are the Haptek Player[1] as well as an application based on Visage[2], both capable of using a TTS engine such as Nuance's RealSpeak™ US English voice which we have used in most applications. Alternatively, audio-only output using Cloudgardens[3] TalkingJava is also presently supported.

3 Modularity

At every step of development we have kept modularity, flexibility and interchangeability of different parts of the system in mind. The context-free grammar used by the TGM is contained in a human readable text-file and is completely language independent. Combined with being able to use more or less any TTS engine, audio output is truly language independent. The OM is furthermore easily replaceable to support different external applications, with all the different output modules being selectable *at runtime*.

The most recent work we have done to explore the abovementioned characteristics has been to add support for the RoboSoccer domain as well as support for the Virtual Storyteller project[4] at the University of Twente. The adaptations necessary to support these domains were completed in 13 and 22 hours, respectively, which serve as support for our claims of modularity and adaptability.

The ability to adapt the system to diverse domains enables it to be used in a multitude of applications which can benefit from an embodied agent conveying situational information.

4 Demonstration

DEIRA is able to run demonstrations of the virtual storyteller, the robosoccer and the horse racing module, with demonstrations of the different instances lasting about 3 minutes per instance. This can be done using a sufficiently powerful laptop which is very preferably connected to external audio and video outputs.

References

[1] F.L.A. Knoppel et al. Trackside DEIRA: A Dynamic Engaging Intelligent Reporter Agent. *In Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 112-119

[1] http://www.haptek.com
[2] http://www.visagetechnologies.com
[3] http://www.cloudgarden.com
[4] http://hmi.ewi.utwente.nl/showcase/The%20Virtual%20Storyteller

Demonstration of Online Auditory Scene Analysis

J.D. Krijnders T.C. Andringa

Auditory cognition group, Rijksuniversiteit Groningen,
P.O.Box 407, 9700 AK Groningen

Abstract

We show an online system for auditory scene analysis. Auditory scene analysis is the analysis of complex sounds as they occur in natural settings. The analysis is based on models of human auditory processing. Our system includes models of the human ear, analysis in tones and pulses and grouping algorithms. This systems forms the basis for several sound recognition tasks, such as aggression detection and vowel recognition.

Current sound recognition systems function well in controlled spaces with one, well defined source. But as soon as multiple sound sources are present the performance drops rapidly[3]. Humans on the other hand seems to have little problems recognising sounds in the presence of noise[5] or other sources. One of the goals of (computational) auditory scene analysis[7] is to try to bridge this gap by basing its methods on what is known about human auditory processing. We will demonstrate our system based on techniques from auditory scene analysis augmented by algorithms that select spectro-temporal areas with positive signal-tonoise ratio.

1 Methods

The model demonstrated is based on a transmission line model of the human ear[1]. Its latency is lower than filterbank implementations[2], which makes it more suitable for online models. The output is squared, leaky-integrated and down-sampled which results in a energy representation. This representation is called a cochleogram. To find areas of the cochleogram where tones or pulses dominate, we filter it with two segment-dependent filters. These filters match the shape of the tone response and the pulse response of the cochleogram up to two standard deviations of white noise under the peak of the response. The result is two representations, one which indicates the pulsality, the other indicating the tonality of each time-frequency combination of the signal. These measures can be interpreted as local signal-to-noise ratios under the assumption that the signal is a pulse, resp. a tone. In both representations the neighboring local maxima are joined to form regions of the spectrum with are likely to be one continuous tone or pulse. These regions we call signal components. Tonal signal components are then grouped into harmonic complexes, this can increase the robustness of the signal components and leads to groups of signal components that are highly likely to stem from a single source.

2 Discussion

The cochlea model is successfully applied in several sound monitoring projects in the Netherlands[6]. The complete system is used in several projects ranging from aggression detection[8] to environmental sounds recognition[4].

3 Demo requirements

The demo runs continuously, but with explanation will take about ten minutes to visit. Requirements would include a large screen or beamer and a table for it.

References

[1] H. Duifhuis, H.W. Hoogstraten, S.M. Netten, R.J. Diependaal, and W. Bialek. *Cochlear Mechanisms: Structure, Function and Models*, chapter Modelling the cochlear partition with coupled Van Der Pol oscilators, pages 395404. Plenum, New York, 1985.

[2] Toshio Irino and Roy Patterson. A time-domain, level-dependent auditory filter: The gammachirp. *Journal of the Acoustical Society of America*, 101(1):412419, January 1997.

[3] Richard P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):115, July 1997.

[4] M.E. Niessen, L. Van Maanen, and T.C. Andringa. Disambiguating sounds through context. In *Proc. IEEE International Conference on Semantic Computing*, 2008.

[5] H.J.M. Steeneken. *On measuring and predicting speech intelligibility*. PhD thesis, University of Amsterdam, 1992.

[6] P.W.J. van Hengel and T.C. Andringa. Verbal aggression detection in complex social environments. In *Advanced Video and Signal Based Surveillance, 2007*. AVSS 2007. IEEE Conference on, pages 1520. IEEE, 2007.

[7] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis*. John Wiley and Sons, Holoken, NJ, 2006.

[8] W. Zajdel, J.D. Krijnders, T.C. Andringa, and D.M. Gavrila. Cassandra: audio-video sensor fusion for aggression detection. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS 2007*, pages 200205, 2007.

A Generic Rule Miner for Geographic Data

Joris Maervoet [a,b] Patrick De Causmaecker [b] Greet Vanden Berghe [a]

[a] *KaHo Sint-Lieven, Departement Industrieel Ingenieur, Vakgroep IT, Gebr. Desmetstraat 1, 9000 Gent, Belgium*
[b] *Katholieke Universiteit Leuven Campus Kortrijk, Faculty of Sciences, Department of Informatics, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium*

Abstract

The GIS and mobile navigation markets are large and still have growth potential. The effectiveness of end user products in these markets is directly related to the quality of the geographic data provided. Likewise, the importance of quality maintenance and related tools increases for digital content providers. This demonstration paper presents a quality maintenance tool for the company Tele Atlas. It is implemented to discover regularities, in the form of rules, and corresponding anomalies in geographic data by the integration of an algorithm from the field of Descriptive Inductive Logic Programming (DILP). It is generic in the sense that it discovers regularities and/or anomalies that hold over a selection of related geographic feature and property types, and apply to a selection of geographic areas.

1 Introduction

The company Tele Atlas is a geographic content provider that collects its data by analysing satellite images, by processing information from other organisations and by processing data that was captured on the field by mobile mapping vehicles, which are shown in Figure 1. Even though each update to the core database is subject to a set of fundamental quality checks, anomalies reside in the geographic data, due to human mistakes or inconsistencies between updates from different sources. More information about the case can be found in [2], in which also a case study with DILP is discussed in further detail.

Figure 1: Geographic data collection with mobile mapping vehicles

We integrated a prototype tool in the company's environment that enables data engineers to automatically mine for regularities and anomalies in the geographic data. It extracts and combines geographic data into higher level concept knowledge, which is fed to the WARMR[1] algorithm. This algorithm finds all the queries that satisfy a given language bias L and cover at least t interpretations of the given set of interpretations E. These frequent queries are processed afterwards into query extensions (rules) describing regularities that apply to the input data. Violations to these rules are the candidate anomalies. Furthermore the tool is able to translate resulting rules into the rule engine language used by the company and to import previously induced knowledge such that the DILP algorithm takes it into account during the search process. Note that the tool can be used with any DILP algorithm that fits into the framework described by Stolle et al. [3].

2 Demonstration

Figure 2 depicts the component diagram of the generic rule miner. The user interaction and outcome of each component will be shown during the demonstration, by use of a set of cases. For example, relations between various properties of road elements are investigated in the region of NW Barcelona, which results in the finding that 99.2% of the road elements with a specific name have importance 6. Next, violations to the LP rule are traced and the rule is exported into the company's format.

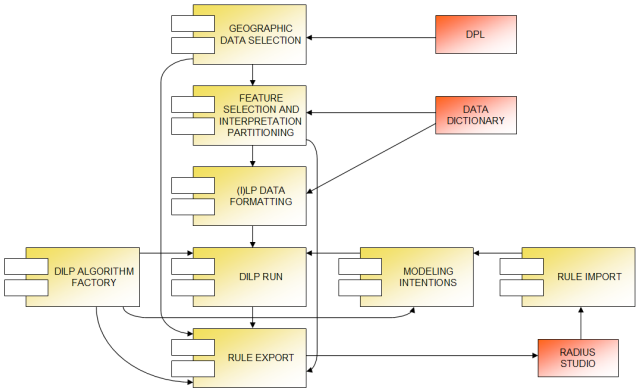


Figure 2: Component diagram of the generic rule miner

The *Geographic Data Selection Component* is controlled by an end user to manually select the regions of interest that correspond to the application area of resulting rules. The *Feature Selection and Interpretation Partitioning Component* allows the user to select a desirable way to partition the data into interpretations using some feature and attribute types of choice. The *(I)LP Data Formatting Component* converts the interpretations into the format of LP and maps the data model on a set of LP predicates. The *Rule Import Component* is able to import already approved rules from the company's knowledge base. The *Modelling Intentions Component* allows the end user to reflect modelling intentions in the DILP language bias and background knowledge. Finally, the *Rule Export Component* collects the context in which the global experiment was set up from several modules to convert resulting rules into the company's rule format.

3 Further specifications

This prototype tool is developed as part of an R&D research project funded by IWT (050730). It is implemented in Java and uses the WARMR implementation of the ACE Datamining System (K.U.Leuven). It interfaces with the APIs provided by Tele Atlas to acquire geographic data (from an Oracle database running on the local system) and the data model. It uses the mProlog libraries to trace back violations. A JAXB library was generated from the company's rule format XSD to enable the exportation of rules. Both the presentation of the case and the software demonstration take 15 minutes. Special thanks to Gert Vervaet (Tele Atlas) for the collaboration and to Katja Verbeeck (KaHo Sint-Lieven) for the suggestions made.

References

[1] Luc Dehaspe. *Frequent Pattern Discovery in First-Order Logic*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1998.

[2] Joris Maervoet, Patrick De Causmaecker, Ann Nowé, and Greet Vanden Berghe. Feasibility study on applying descriptive ILP to large geographic databases. Working Papers JM08061, KaHo Sint-Lieven, Gent, June 2008.

[3] Christian Stolle, Andreas Karwath, and Luc De Raedt. CLASSIC'CL: An integrated ILP system. In *Proceedings of the 8th International Conference of Discovery Science*, pages 354–362. Springer-Verlag, 2005.

Face Finder

Thomas Mensink[a] Jakob Verbeek[b]

[a]*SNN, Radboud Universiteit, Nijmegen*, [b]*LEAR team, INRIA Rhône-Alpes, Grenoble*
T.Mensink@science.ru.nl, Jakob.Verbeek@inria.fr

Abstract

We demonstrate a high-level search system that allows users to find images of people in a database of captioned news images. The system does not require a labeled training set with example faces of each person; instead the system is based on the weak supervision provided by the captions. Because the system does not rely on supervised face recognition techniques, it is possible to use it in an open-ended setting. Given the name of the queried person, the captions are scanned for this name to yield an initial set of candidate images. These images are further processed by applying a face detector, and then the visual analysis of the detected faces is used to filter out the queried person in an unsupervised manner. For recall levels up to 50% to 60% (which are sufficient to fill the first few result pages) we achieve a precision of 90% averaged over an evaluation set of 23 queries for different people.

1 Introduction

Automatic analysis of online published news streams is important as they are an increasingly important source to fulfill the information need of people, and they publish news articles at a high frequency. Identification of faces of a certain person in news photographs is a challenging task, significantly more so than recognition in the usual controlled setting of face recognition: we have to deal with imperfect face detection and alignment procedures, and also with great changes in pose, expression, and lighting conditions, and poor image resolution and quality. Our demonstration shows how to answer queries for images depicting specific people in databases of captioned news photographs.

When searching for faces of a certain person, a simple system could (i) query the database for captions containing the the name, and (ii) rank or filter the result images by the confidence level of a face detector to retain only images that contain faces. An example of such a system is IDIAP's Google Portrait[1] and a also Google's "advanced image search"[2] has a similar feature.

Although such a system correctly rejects images without faces (or at least without detections), the performance of such a system is clearly limited by the fact that it returns all images with detected faces, not only those depicting the queried person; thus leading to a high false positive rate.

Perhaps naively, one could think of training a face recognition system for all people that appear in the database in question. However, for a system using a large database it is completely infeasible to train a supervised recognition system for all people we could potentially query for. Nevertheless, we could use the information contained in captions.

Recently research [1] has shown that initial text-based results can be significantly improved by filtering images based on analysis of the detected faces using visual features. While using the caption alone leads to a disappointing precision of 44% (fraction of faces belonging to the queried person among all returned faces, averaged over queries for 23 people), adding face analysis increases average precision to 71%, at a recall of 85%. The main failure of this work is in cases where the queried person's face is relatively rare: when under say 40% of the detected faces represents the queried person the precision may be very low (under 50%).

Recently we have shown [2] that using a form of 'query expansion' results can be significantly improved. The set of news stories found by querying the captions with a name, is extended by querying the database for names that appear frequently together with the queried person, we refer to those names as 'friends'. We

[1] http://www.idiap.ch/googleportrait
[2] http://images.google.com

use the 'query expansion' —the set of faces found in images with friends appearing in the caption, but not the queried person— to obtain a notion of whom we are *not* looking for.

2 Face Filtering using a Gaussian Mixture Model

We apply the query expansion idea to a generative mixture model to filter the text-based results. While we have shown that a linear discriminant method slightly outperforms the generative mixture model, the former is much faster to learn and allows it to be used in a real-time system. A Gaussian mixture model can be used by framing the problem to a constrained clustering problem. We define a mixture with two components, one to model the queried person (foreground) and one to model the background. We fix the parameters of the background Gaussian, thus only the parameters of the foreground Gaussian have to be estimated.

We associate an unknown assignment variable $\gamma \in \{0, \ldots, F\}$ with each document in the query set, to represent which, if any, of the F faces in the image belongs to the queried person, i.e. the foreground component. We define a mixture model over the features of the detected faces $\mathcal{F} = \{f_1, \ldots, f_F\}$, marginalizing over the unknown variable γ (1-2). Which component is used in $p(f_i|\gamma)$ depends on the value of γ, the foreground component is only used if $i = \gamma$ (3).

$$p(\mathcal{F}) = \sum_{\gamma=0}^{F} p(\gamma) p(\mathcal{F}|\gamma) \tag{1}$$

$$p(\mathcal{F}|\gamma) = \prod_{i=1}^{F} p(f_i|\gamma) \tag{2}$$

$$p(f_i|\gamma) = \begin{cases} \mathcal{N}(f_i; \mu_{\text{BG}}, \Sigma_{\text{BG}}) & \text{if } \gamma \neq i \\ \mathcal{N}(f_i; \mu_{\text{FG}}, \Sigma_{\text{FG}}) & \text{if } \gamma = i \end{cases} \tag{3}$$

We use the EM algorithm to maximize the log-likelihood of the set of faces in all documents with respect to the parameters of the foreground component. After optimization we use the maximum likelihood assignment to determine which, if any, face represents the queried person. This model implements the constraint that only one face per document may be selected.

In the case of query expansion, we use a more elaborated background component. Instead of a single Gaussian representing all background faces, we use a Mixture of Gaussians, each representing a face from the query expansion and one for the other background faces. The Gaussians for the faces from the query expansion are learned with the method described above. As before during learning the foreground parameters we fix the background model, and only the parameters of the foreground model are optimized.

3 Demonstration

We present a web-based search interface to a large collection of news photos with captions. The system is able to find quickly images that contain the face of the queried person. We show the results of (i) filtering faces after text-based queries, and (ii) using our system based on query expansion. Face Finder is developed by the authors at the Learning and Recognition in Vision (LEAR) team of INRIA Rhône Alpes, Grenoble, France. It was part of an research internship supported by VSBfonds and the EU funded project CLASS (Cognitive-Level Annotation using latent Statistical Structure, `http://class.inrialpes.fr/`). The system runs on a standard laptop, and the demonstration takes about 15 minutes.

The demo of the generative model with query expansion on the Yahoo! News data set is also available at `http://lear.inrialpes.fr/~verbeek/facefinder`.

References

[1] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Automatic face naming with caption-based supervision. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[2] Thomas Mensink and Jakob Verbeek. Improving people search using query expansions: How friends help to find people. In *10th European Conference on Computer Vision*, 2008.

OperettA: A Prototype Tool for the Design, Analysis and Development of Multi-agent Organizations

Daniel OKOUYA[a] Virginia Dignum[b]

[a] *Dept. Information and Computing Sciences, Utrecht University*
3508 TB Utrecht, The Netherlands

Abstract

OperettA is a graphical tool that supports the design, verification and simulation of OperA models. It ensures consistency between different design parts, provides a formal specification of the organization model, and is prepared to generate a simulation of the application domain.

1 Introduction

The OperettA is an IDE (Integrated Development Environment) developed to support the design, analysis and development of agent organizations using the OPERA methodology [1]. It is intended to support software engineers and developers in both developing and documenting the various aspects of specifying and designing a multi-agent organization

The OperA model proposes an expressive way for defining open organizations that explicitly distinguishes between the organizational model and the agents who will act in it. That is, OperA enables the specification of organizational requirements and objectives, and at the same time allows participants to have the freedom to act according to their own capabilities and demands. The OperA framework consists of three interrelated models. The **organizational model (OM)** is the result of the observation and analysis of the domain and describes the desired behavior of the organization, as determined by the organizational stakeholders in terms of objectives, norms, roles, interactions and ontologies. The **social model (SM)** maps organizational roles to specific agents. Agreements concerning the role(s) an agent will play and the conditions of the participation are described in social contracts. The **interaction model (IM)** specifies the interaction agreements between role-enacting agents as interaction contracts.

The Electronic Institution Development Environment (EIDE) for ISLANDER which results in an AMELI [2] implementation can be seen as similar to OperettA. However, EIDE is developed among other things for the specification of fully regimented institutions, and as such does not meet the OperA's requirements of internal autonomy and collaboration autonomy [1]. OperettA has been implemented following the model driven software development paradigm, which enables the introduction and combination of different formal methods hence enabling the modeling activity through systematic advices and model design consistency checking.

2 The OperettA Architecture

The Operetta prototype is implemented using MetaEdit+1, a generic customizable model driven software development environment suitable for prototyping. The prototype incorporates Racer DL2 reasoning

[1] http://www.metacase.com/
[2] http://www.sts.tu-harburg.de/~r.f.moeller/racer/

system, SWI-prolog interpreter3, MCMAS4 model checker and Brahms5 as a possible simulation environment. The OperettA Architecture, which is depicted in Figure 1, is further described below.

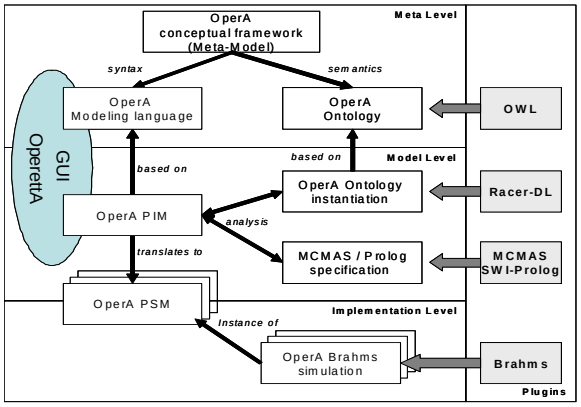


Figure 1 - The OperettA conceptual Architecture

Following the model driven paradigm, OperettA consists of 3 different levels. The **Meta Level** is directly based on the OperA conceptual framework and provides its syntax and semantic specifications. Syntax is derived from the OperA BNF and semantics are defined as an OWL[6] ontology[i]. At the **Model Level** the development environment for OperA OM specifications is defined. It provides a multi-viewed GUI (graphical user interface) and model verification support. At this level, a Platform Independent Model (PIM) is constructed for the organization which is semantically checked trough OWL. Finally, the **Implementation Level** (under construction) enables the generation of Platform Specific Models (PSM). Brahms is currently used as a simulation environment for organizations in which normative properties of the organization can be verified for different populations with emergent behavior.

3 Conclusion and Future Works

OperettA with its underlying model driven architecture that take into account semantic provides interesting features that are extremely useful in the modeling of complex environments that require the integration of organizational structures and individual (emergent) behaviors. We are currently working on the integration of OperA and Brahms for the specification and evaluation of integrated models [3]. In the future, other implementation platforms (such as Repast) will be supported, and the tool will be integrated in an OperA open environment that enables the participation of heterogeneous agents in the same organization

References

[1] V. Dignum, F. Dignum, J.J. Meyer (2004): An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations. *Knowledge Engineering Review*, Cambridge University Press, 19(2), pp. 147-174, 2004.

[2] Esteva, M., Rosell, B., Rodriguez-Aguilar, J. A., and Arcos, J. L. (2004). AMELI: An Agent-Based Middleware for Electronic Institutions. In: *Proc. AAMAS'04*. 236-243.

[3] B.J. van Putten, V. Dignum, M. Sierhuis, S. Wolfe (2008): Integrating Organizational and Emergent Views on Agent-Based Modeling. *Submitted*.

[3] http://www.swi-prolog.org/
[4] http://sse.cs.ucl.ac.uk/projects/mcmas/
[5] http://www.agentisolutions.com/
[6] http://www.w3.org/TR/owl-ref/

Browsing and Searching the Spoken Words of Buchenwald Survivors

Roeland Ordelman Willemijn Heeren Arjan van Hessen Djoerd Hiemstra
Hendri Hondorp Marijn Huijbregts Franciska de Jong Thijs Verschoor

Human Media Interaction
University of Twente, Enschede, The Netherlands

1 The Buchenwald demonstrator

The 'Buchenwald' project is the successor of the 'Radio Oranje' project that aimed at the transformation of a set of World War II related mono-media documents –speeches of the Dutch Queen Wilhelmina, textual transcripts of the speeches, and a database of WWII related photographs– to an attractive online multimedia presentation of the Queen's speeches with keyword search functionality [6, 3]. The 'Buchenwald' project links up and extends the 'Radio Oranje' approach. The goal in the project was to develop a Dutch multimedia information portal on World War II concentration camp Buchenwald[1]. The portal holds both textual information sources and a video collection of testimonies from 38 Dutch camp survivors with durations between a half and two and a half hours. For each interview, an elaborate description, a speaker profile and a short summary are available.

The first phase of the project was dedicated to the development of an online browse and search application for the disclosure of the testimonies. In addition to the traditional way of supporting access via search in descriptive metadata at the level of an entire interview, automatic analysis of the spoken content using speech and language technology also provides access to the video collection at the level of words and fragments. Research in this phase was dedicated to the automatic annotation of the interviews using speech recognition technology [5] and combining manual metadata per interview with the within-interview automatic annotations for retrieval of both entire interviews and interview fragments, given a user's query [4]. Moreover, having such an application running in the public domain allows us to investigate other aspects of user behavior next to those investigated in controlled laboratory experiments.

The second stage of the project aims at (i) the optimization of the automatic annotation procedure [1], (ii) interrelating all available multimedia resources, such as written summaries to exact video locations or locations to maps or floor plans [2], and connected to this, (iii) further development of a user interface that allows for the presentation of this information given the various user needs.

While survivors of World War II can still personally tell their stories, interview projects are collecting their memories for generations to come. Such interview collections form an increasingly important addition to history documented in written form or in the form of artifacts. Whereas social scientists and historians typically annotate interviews by making elaborate summaries or sometimes even full transcripts, by assigning keywords from thesauri and by establishing speaker profiles, catalogs based on these manually generated metadata do not often contain links into video documents. That is, they do not support retrieval of video fragments in response to users' search queries; results are typically entire videos - that may be hours long - or (parts of) the transcripts.

The interview browse and search application of the 'Buchenwald' portal shows a multimedia search application based on both the conventional, manual metadata as well as automatic speech recognition output. It is part of a website on Buchenwald maintained by the Netherlands Institute for War Documentation (NIOD) that gives its user a complete picture of the camp then and now by presenting written articles, photos and the interview collection.

[1] Buchenwald: `http://www.buchenwald.nl`

After the audio tracks had been separated from the video documents, the audio was processed by the open source speech recognition toolkit SHoUT[2] developed at the University of Twente, resulting in coherent speaker segments and a time-stamped transcript for indexing. For retrieval, the open source XML search system PF/Tijah is being used[3].

The user interface supports browsing and search in the collection. To start browsing the collection, a user can request a list of all available videos. Each result contains links to the short summary, the speaker's profile, the elaborate description and the video document (Figure 1). To search the collection, a standard text search field is provided. If results are found, they are listed in the same format as the browse list, with the alteration that two types of results are available: interview results and fragment results.

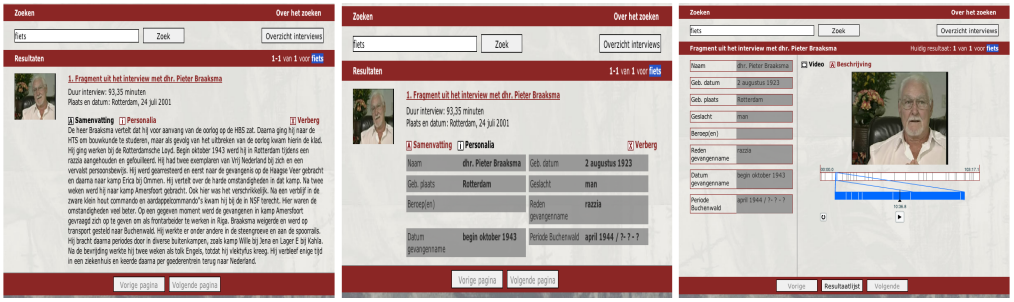


Figure 1: Screen shots of the result list, showing the short summary, the speaker's profile and the video browser

Interview results imply hits in the textual, manual metadata, and fragment results imply hits in the speech recognition output. In the former case, the terms matching the user's query are highlighted in color in the textual metadata. In the latter, the video link directs the user to the exact speaker segment that contains the hit.

References

[1] F.M.G. de Jong, D. Oard, R.J.F. Ordelman, and S. Raaijmakers. Searching spontaneous conversational speech. *SIGIR Forum*, 41(2):104–108, 2007. ISSN=0163-5840.

[2] F.M.G. de Jong, R.J.F. Ordelman, and M.A.H. Huijbregts. Automated speech and audio analysis for semantic access to multimedia. In *Proceedings of Semantic and Digital Media Technologies, SAMT 2006*, volume 4306 of *Lecture Notes in Computer Science*, pages 226–240, Berlin, 2006. Springer Verlag. ISBN=3-540-49335-2.

[3] W.F.L. Heeren, L.B. van der Werff, R.J.F. Ordelman, A.J. van Hessen, and F.M.G. de Jong. Radio oranje: Searching the queen's speech(es). In C.L.A. Clarke, N. Fuhr, N. Kando, W. Kraaij, and A. de Vries, editors, *Proceedings of the 30th ACM SIGIR*, pages 903–903, New York, 2007. ACM.

[4] Djoerd Hiemstra, Roeland Ordelman, Robin Aly, Laurens van der Werff, and Franciska de Jong. Speech retrieval experiments using xml information retrieval. In *Proceedings of the Cross-language Evaluation Forum (CLEF)*, 2007.

[5] M.A.H. Huijbregts, R.J.F. Ordelman, and F.M.G. de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of SAMT 2007*, volume 4816 of *Lecture Notes in Computer Science*, pages 78–90, Berlin, 2007. Springer Verlag.

[6] R.J.F. Ordelman, F.M.G. de Jong, and W.F.L. Heeren. Exploration of audiovisual heritage using audio indexing technology. In L. Bordoni, A. Krueger, and M. Zancanaro, editors, *Proceedings of the first workshop on intelligent technologies for cultural heritage exploitation*, pages 36–39, Trento, 2006. Universit di Trento. ISBN=not assigned.

[2] SHoUT ASR toolkit: `http://wwwhome.cs.utwente.nl/~huijbreg/shout/`
[3] PF/Tijah: `http://dbappl.cs.utwente.nl/pftijah/Main/HomePage`

Temporal Interaction between an Artificial Orchestra Conductor and Human Musicians

Dennis Reidsma Anton Nijholt Pieter Bos

University of Twente, Human Media Interaction
Faculty EEMCS, University of Twente, Enschede, the Netherlands.

1 Introduction

One interest of our research group is in (1) developing Ambient Entertainment technologies and applications that (2) interact in a coordinated way with human partners using (3) a multitude of different sensors observing many characteristics of the partner and using (4) many different channels of expression (such as sound, visuals, speech and embodiment with gestures and facial expressions). The Virtual Conductor project [1, 2] concerns the development of the first properly interactive virtual orchestra conductor that can conduct a piece of music through interaction with musicians, leading and following them while they are playing. Its observations consist of different manners of musical processing of the incoming signal from a microphone. The forms of its expressions are defined by the possible conducting patterns and by the timing, speed, amplitude and smoothness with which those patterns are expressed. The interaction is focused on the tempo of the musicians and includes a correction module that interactively corrects the tempo when the musicians are playing too slow or too fast.

We describe our motivation for developing such a system; related work in the areas of ambient entertainment and coordinated timing, automatic music processing, virtual humans and conducting; the design, implementation and evaluation of the Virtual Conductor; and, finally, contains a discussion of the resulting system, general applicability of the ideas and technology developed in this project, and expected developments in the (ongoing) Virtual Conductor project.

2 Motivation

More and more, music is becoming a theme in computer based entertainment. Games in which interaction with or through music plays a central role are on the rise (see, for example, games such as Guitar Hero, Dance Dance Revolution, Donkey Konga and many, many more). However, for many of those games the interaction through music is mostly one-way: the player must follow a rhythm or riff presented by the computer to achieve a set goal. When a group of people make music, interaction is inherently simultaneous and two-way. Both partners in a musical cooperation are alert to what the other is doing and adapt their own performance to mesh.

In the Virtual Conductor project presented in this paper a major element is the mutual interaction between system and musician with respect to tempo and timing. Some ideas concerning temporal coordination in interaction have been worked out preliminarily in [3], in relation to topics such as synchrony. Here we just note the strong positive relation found in literature between synchrony and positive affect, or between synchrony and a positive evaluation of the interaction, in human-human interaction, but also in human-computer interaction. Given the literature it seems a reasonable assumption that implementation of modules for synchrony can add to the enjoyment and engagement of users of computational entertainment applications. The Virtual Conductor can be seen as one of the first ambient entertainment applications that takes a step in the direction of harnessing interactional synchrony for improving the enjoyment and engagement of the user.

A Virtual Conductor system can be used in several ways. An edutainment application of such technology could be in teaching student conductors. As a reflective tool, the system could show good examples as

well as examples of typical conducting mistakes or allow the student conductor to visualise different ways of conducting a passage to see what it looks like. In combination with the complement of this artificial conductor, namely an artificial orchestra such as the one on display in the Vienna House of Music, a system could be envisioned that detects the student's mistakes and graphically shows them to the student in combination with a better way of conducting. We can also envision this conductor developed further as a rehearsal conductor. The time in which a human conductor can work with a certain ensemble is often limited; if a Virtual Conductor could be used to rehearse the more technical aspects of a piece of music this would leave the human conductor more time to work on the creative and expressive musical aspects of a performance. Finally, a Virtual Conductor could also be made available through the internet to provide the casually interested layman with easy and engaging access to knowledge about, and some do-it-yourself experimentation with, conducting.

3 Survey and Conclusions

A Virtual Conductor has been researched, designed and implemented that can conduct human musicians in a live performance. The conductor can lead musicians through tempo, dynamics and meter changes. During test sessions the musicians reacted to the gestures of the conductor. It can interact with musicians in a basic way, correcting their tempo gracefully when they start playing faster or slower than is intended. Feedback from the musicians who participated in the tests shows that the musicians enjoy playing with the virtual conductor and can see many uses for it, for example as a rehearsal conductor when a human conductor is not available, or a conductor for playing along with when practicing at home.

Several audio algorithms have been implemented for tracking the music as played by the ensemble. Among those, the beat detector can track the tempo of musicians and the score follower can track where musicians are in a score, in real-time. The possibilities of these audio algorithms reach further than what is currently used for feedback in the Virtual Conductor and will be very useful for future extensions of the system.

While this article presents the results of the first stages of the Virtual Conductor project, the system is still being extended in several directions. We have been working on a detailed analysis of a corpus of performances and rehearsals that we recorded on video, in order to find out more about the intentions and signals used in conducting and about the structure of rehearsals. The repertoire of expressions that the Virtual Conductor can use is continuously being extended. We are working on implementing rehearsal modules that allow the Virtual Conductor to plan a structured series of rehearsal sessions, adapting later sessions in response to the progress achieved in the earlier sessions. Also, the planning and scheduling modules used to produce the conducting animations have been extensively redesigned in order to allow a much more flexible and responsive behavior generation that can handle several expressive channels at the same time. This will allow us to combine the basic beat patterns with an extensive repertoire of left hand gestures, facial expressions, gaze behavior and other non verbal expressions.

References

[1] D. Reidsma, A. Nijholt, A., and P. Bos. Temporal Interaction between an Artificial Orchestra Conductor and Human Musicians. ACM Journal on Computers and Entertainment, accepted for publication.

[2] M. ter Maat, R. Ebbers, D. Reidsma and A. Nijholt. Beyond the beat: modelling intentions in a virtual conductor. In: *Proceedings of the 2nd International Conference on intelligent Technologies For interactive Entertainment (INTETAIN)*. ACM Digital Libraries, 1-10.

[3] A. Nijholt, D. Reidsma, H. van Welbergen, H.J.A. op den Akker, and Z.M. Ruttkay. Mutually Coordinated Anticipatory Multimodal Interaction. In: *Nonverbal Features of Human-Human and Human-Machine Interaction*, 29-31 October 2007, Patras, Greece. Lecture Notes in Computer Science 5042, Springer Verlag, Berlin, 2008, 73-93.

Emotionally Aware Automated Portrait Painting

Michel Valstar Simon Colton Maja Pantic

Department of Computing, Imperial College London,
180 Queen's gate, London SW7 2AZ, England
{*Michel.Valstar, S.Colton, M.Pantic*}*@imperial.ac.uk*

Abstract

We propose to demonstrate The Emotionally Aware Painting Fool, a novel system that combines a machine vision system able to recognise emotions with a non-photorealistic rendering (NPR) system to automatically produce portraits of the sitter in an emotionally enhanced style. The vision system first records a short video clip of a person showing a basic emotion. The system then analyses this video clip, locating and tracking facial features, recognising the basic emotion shown and when this emotion was strongest. This information is then passed to the NPR software. The detected emotion is used to choose appropriate (simulated) art materials, colour palettes, abstraction methods and painting styles, so that the rendered image may heighten the emotion being expressed. The live demonstration shows how each element of The Emotionally Aware Painting Fool functions and produces a portrait in approximately 7 minutes.

1 Introduction

We are interested in the notion of computational creativity, in particular the question: under what circumstances (if any) is it appropriate to describe the behaviour of a computational system as creative. Visual art is a domain where human creativity flourishes, so Non-Photorealistic Rendering (NPR) – where art materials and artistic styles are simulated – would appear to be an ideal domain in which to test computational models of creativity. Unfortunately, however, NPR researchers have tended to eschew the potential for software to act as creative collaborators in art projects, opting instead to build systems which merely enhance the efficiency/creativity of users. In fact, due to its close relationship with human creativity, some authors seem almost apologetic about simulating artistic techniques.

We disagree with this assessment, and we are currently building an automated painter (called The Emotionally Aware Painting Fool) which we hope will eventually be accepted as a creative artist in its own right. We have taken the approach of identifying what appear to be some necessary high-level conditions for creative behaviour. One such condition is that the software exhibits appreciation in its behaviour, and we describe here how we have addressed the issues of the software appreciating both its subject matter and the way its rendering choices affect the picture it produces. To start addressing these issues, we added an expert system to The Painting Fool which takes a description of the sitter's facial expression, and chooses from an extensive range of abstraction, colouring and rendering methods which – taken together – specify an artistic style appropriate to the emotion shown.

The demonstration was build initially for the British Computing Society's Machine Intelligence Award 2007, which the authors won. The demonstration runs on a laptop, with the video being captured by a small webcam, and a projector used to show the progress of the system on a screen.The demonstration takes about 7 minutes for each portrait painted and is intended for the general public: no technical knowledge is required from the sitters. The system on which this demonstration is based has been described in detail in [1].

2 Facial expression analysis

We used a fully automated system [3] that can recognise the six basic emotions: anger, disgust, fear, happiness, sadness or surprise. The system uses features based on the geometric relations between fiducial facial

Figure 1: Example portraits. Emotions expressed, from left to right: fear, happiness, disgust, anger, sadness, and surprise.

feature points. Using a combination of GentleBoost feature selection and Support Vector Machine classification we detect the activation of facial muscle groups, so-called Action Units (AUs, [2]). Ekman et al. have shown that it is straightforward to recognise which expression of emotion was displayed if we know what AUs were activated during that expression [2]. In our approach we use neural networks to map the detected AUs to an emotion.

Recently, we have proposed a system that is capable of analysing both the morphology of an expression (i.e. determine which AUs were present) as well as the temporal dynamics of an expression [3]. For each AU we find exactly when the facial action starts, when it reaches its peak, when it starts to return to neutral and when it has returned to its neutral phase. We use this knowledge to find the frame in a video in which the emotion is displayed strongest. This frame is then sent to The Painting Fool, together with information about which emotion was recognised and the location of the facial points in that apex frame.

3 The Painting Fool NPR System

The NPR system is given a digital image which may or may not have been annotated with the boundaries of scene-elements within the image (for instance, the user might choose to provide details of where the eyes, nose and mouth of a person are in a digital image). It produces an artistic rendering of the image in a two-stage process. Firstly, it segments the entire image and separately segments the scene-elements producing a list of segmentations. Secondly, it takes each shape in each segmentation and renders it with simulated art materials such as acrylic paints, pastels, pencils, etc. We do not claim that our NPR techniques are particularly novel. However, we believe that by applying the NPR techniques that are selected by the expert system to be appropriate to the emotion shown, we can enhance the emotional expression of a portrait.

4 Emotionally Aware Portraits

To produce the most impressive demonstration, we opted for full automation, with the demonstrators having only to start and stop a video camera into which subjects were asked to express an emotion, for example by smiling. The Emotionally Aware Painting Fool produces its pictures live, i.e., it renders each stroke in real time, which can add value to the demonstration. The full rendering process can take minutes or hours, so we chose artistic styles for each of the emotions which would complete the picture in around three minutes. In Fig. 1, we present some example portraits for each emotion. More examples can be found on the website www.thepaintingfool.com.

References

[1] S. Colton, M.F. Valstar, and M. Pantic. Emotionally aware automated portrait painting. In *Proc. Int'l Conf. Dig. Interact. Media in Entert. & Arts*, 2008. Submitted for peer review.

[2] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System*. A Human Face, 2002. Salt Lake City.

[3] M. F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *IEEE W'shop Human Computer Interaction (HCI'07), IEEE ICCV'07*, 2007.

Demonstration of a Multi-agent Simulation of the Impact of Culture on International Trade

Tim Verwaart John Wolters

LEI Wageningen UR, Postbus 29703, 2502 LS den Haag.

Abstract

A multi-agent model of trade in mono-cultural and multi-cultural setting will be demonstrated. The model allows for configuration of cultural background in individual agents, according to the Hofstede's five dimensions of national culture. The model is built as a tool for social-scientific research into the efficiency international trade and supply chain formation, in particular in a context of institutional economics. The emergence of different trade patterns under different cultural settings will be demonstrated.

1 Theoretical background

A series of papers by G.J. Hofstede et al. [1, 2, 3, 4, 5] describe the differentiation of behavior of individuals in trade processes according to the individual's cultural background. The papers are based on the work of G. Hofstede [6], who found that national cultures can be characterized according to five dimensions:

- the extent to which the less powerful in a society expect and accept that power is distributed unequally (power distance, or hierarchical versus egalitarian cultures);
- the extent to which the members of a society can accept that certain things are unknown and that rules for behavior are ambiguous (uncertainty avoiding versus uncertainty tolerant cultures);
- the extent to which the members of a society feel to be individuals responsible for their personal interest or group members responsible for common interest (individualism versus collectivism);
- the extent to which the members of a society are oriented toward performance and competition or toward care-taking and co-operation (masculine versus feminine cultures);
- the extent to which the members of a society pursue status, consumption, and immediate profit, or pragmatically pursue long-run goals and virtues (short-term versus long-term orientation).

The papers by G.J. Hofstede et al. [1, 2, 3, 4, 5] define rules for culturally differentiated behavior of trading agents in the processes displayed in figure 1. The purpose of these models of culture is to advance the understanding of international trade processes and supply chain formation in different mono-cultural and multi-cultural settings, in a context of institutional economics, and to compare the efficiency of institutions. The research method in which the models are applied, combines multi-agent simulation and human gaming simulation, as indicted in figure 2. For examples of such research, see [7, 8].

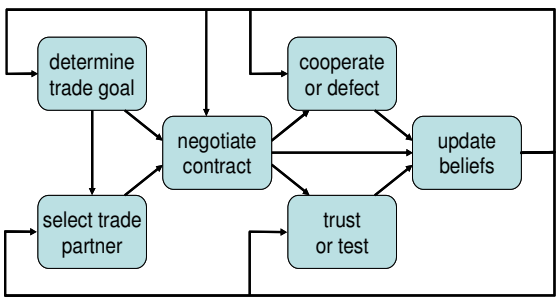


Figure 1. Process model of trading agents

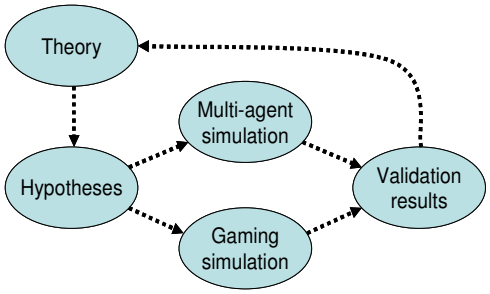


Figure 2. Research cycle

2 Multi-agent Simulation

The simulation application allows for configuring individual agents with cultural parameters, representing the Hofstede dimensions, and some personal traits, like impatience, and negotiation preferences and parameters. Agents have a role of either supplier or customers. They trade a commodity that has a hidden quality attribute, so the suppliers have an opportunity to cheat the customers. The customers can either trust the suppliers, or put their purchases to the test with the tracing agent, that will at the cost of a fee inform them about the real quality, and fine the supplier in case of deceit. A number of suppliers and a number of customers can be configured for a simulation run, with all customers connected to all suppliers. They may freely select a partner and send a trade proposal, and their partners are free to respond or ignore a proposal. Agents can enter into negotiations with only one partner concurrently. They may come to an agreement, or break-off negotiations if they feel there is insufficient progress or the partner makes unrealistic proposals. They may simple not respond any longer to partner's proposals, in which case the partner will start searching for a new partner after waiting for a while. Thus trade patterns emerge, of which some important observables, to be made at group level or for individual relations, are:

- the number of successful transactions,
- the average quality of commodities traded,
- the number of tracing requests,
- the number of revealed/unrevealed deceits,
- the number of failed negotiations,
- average duration of negotiations.

The application that will be demonstrated, has been implemented in Cormas (http://cormas.cirad.fr/indexeng.htm). The demonstration will show the emergence of patterns of some observables in different settings.

References

[1] G.J. Hofstede, C.M. Jonker, S. Meijer, T. Verwaart. Modelling Trade and Trust across Cultures. *Proceedings of the 4th International Conference on Trust Management, iTrust 2006*, *LNCS* 3986, pp. 120-134, 2006.

[2] G.J. Hofstede, C.M Jonker, T. Verwaart. Uncertainty Avoidance in Trade. *Proceedings of 2008 Agent-Directed Simulation Symposium (ADS'08), Ottawa, April 14-17*, SCS, San Diego: 143-152, 2008.

[3] G.J.Hofstede, C.M. Jonker, T. Verwaart. Modelling Power Distance in Trade. *9th International Workshop on Multi-Agent-Based Simulation at AAMAS'08, May 12-13, Estoril, Portugal*, 2008.

[4] G.J. Hofstede, C.M. Jonker, T. Verwaart. Individualism and collectivism in trade agents. In: Nguyen, N.T. et al., editors, *New Frontiers in Applied Artificial Intelligence, Proceedings of IEA/AIE 2008*, *LNAI* 5027: 492-501, 2008.

[5] G.J. Hofstede, C.M. Jonker, T. Verwaart. Long-term Orientation in Trade. In: K. Schredelseker and F. Hauser, editors, *Complexity and Artificial Markets*, Springer-Verlag, Berlin Heidelberg, Lecture Notes in Economics and Mathematical Systems 614, 2008.

[6] G. Hofstede. *Culture's Consequences, second edition*. Sage Publications, Thousand Oaks, 2001.

[7] S. Meijer, G.J. Hofstede, G. Beers, S.W.F. Omta. Trust and Tracing game: learning about transactions and embeddedness in a trade network, *Production Planning and Control*, 17: 569-583, 2006.

[8] C.M. Jonker, S. Meijer, D. Tykhonov, T. Verwaart. Agent-based Simulation of the Trust and Tracing Game for Supply Chains and Networks, *Journal of Artificial Societies and Social Simulation* 11(3), June 2008.

List of authors