

Designing Generic and Efficient Negotiation Strategies

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 7 juni 2010 om 12:30 uur
door

Dmytro TYKHONOV

Master of Science Artificial Intelligence
geboren te Kharkiv, Oekraïne

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. C. M. Jonker

Copromotor: Dr. K. Hindriks

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. C. M. Jonker,	Technische Universiteit Delft, promotor
Dr. K. Hindriks,	Technische Universiteit Delft, copromotor
Prof. dr. C. Witteveen,	Technische Universiteit Delft
Prof. dr. F. Brazier,	Technische Universiteit Delft
Prof. dr. J. Treur,	Vrije Universiteit Amsterdam
Prof. dr. S. Kraus,	Bar-Ilan University
Prof. dr. C. Sierra,	Thr Artificial Intelligence Research Institute of the Spanish Research Council



SIKS Dissertation Series No. 2010-24

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 978-94-90818-02-9

Copyright ©2010, Dmytro Tykhonov. All rights reserved.

Contents

Contents	iv
1 Introduction	1
1.1 Overview	1
1.2 Negotiation: the basics	5
1.3 Research approach	7
1.4 Negotiation: outcome analysis	8
1.5 Negotiation: knowledge counts	9
1.6 Negotiation Strategies and Tactics	11
1.7 Contribution of the Research	12
1.8 Thesis Overview	12
1.8.1 An Analytic Framework of Negotiation Dynamics and Strategies .	13
1.8.2 An Open Negotiation Architecture for Heterogeneous Agents and A Negotiation Testbed	14
1.8.3 Learning Opponent's Preferences in Negotiation	15
1.8.4 Negotiation Strategy	16
1.8.5 Approximating the Qualitative Vickrey Auction by a Negotiation Protocol	16
1.8.6 Eliminating Issue Dependencies in Negotiation Domains	17
1.9 Publications related to each chapter	18
2 Let's Dance! An Analytic Framework of Negotiation Dynamics and Strategies	23
2.1 Introduction	23
2.2 Related Work	25
2.2.1 Optimal Solutions and Performance Metrics	25
2.3 Negotiation Dance Literature	27
2.4 Negotiation factors	28
2.4.1 Negotiation domains and preference profiles	29
2.4.2 Negotiation strategies	29
2.5 Negotiation Dynamics	30
2.6 Step-Wise Analysis	31
2.6.1 Negotiation Moves	31
2.6.2 Step Metrics and Pattern Properties	33
2.6.3 Sensitivity to Opponent's Preferences	34

2.7	Negotiation Analysis Methodology	36
2.7.1	Phases of the Negotiation Analysis Method	37
2.8	DANS Applied	38
2.8.1	Outcome Analysis	39
2.8.2	Sensitivity to Opponent's Behaviour	40
2.8.3	Sensitivity to Opponent's Preferences	42
2.8.4	DANS analysis results	43
2.9	Conclusions	44
3	A Multi-Agent Environment for Negotiation	49
3.1	Introduction	49
3.2	Application Domain	50
3.2.1	The Added Value of the MAS Paradigm	51
3.2.2	Design Methods Used	52
3.2.3	User Interaction	54
3.3	Agents	56
3.3.1	Agent Architecture	56
3.3.2	State of the Art Negotiating Agents	58
3.4	Multi-Agent System	63
3.4.1	Organisation	63
3.4.2	Interaction	63
3.4.3	MAS Environment	65
3.5	Execution Platform	66
3.6	Results	67
3.6.1	Experimental Setup	67
3.6.2	Experimental Results	69
3.6.3	Approximating Auction Mechanism with Negotiation	74
3.7	Conclusion	75
4	BOP: an Effective Bayesian Learning Algorithm for Opponent Preferences in Negotiation	79
4.1	Introduction	79
4.2	Related Work	82
4.2.1	Learning Approaches	82
4.2.2	Learning in Negotiation	84
4.3	Motivation for Design Choices	86
4.3.1	Structure of the Opponent's Preferences	87
4.3.2	Model of Opponent's Strategy	89
4.4	Learning Approach	91
4.4.1	Bayesian Learning Algorithm for Opponent Preferences	91
4.4.2	Scalable Learning Algorithm	92
4.4.3	Updating Probabilities of Hypotheses	95
4.5	Quality Assessment Method	96
4.5.1	Quality Measures	97

4.5.2	Negotiation Domains and Profiles	99
4.6	Experimental Setup	101
4.7	Experimental Results	104
4.8	Conclusions	113
5	The Benefits of Opponent Models in Negotiation	119
5.1	Introduction	119
5.2	Related Work	120
5.3	Negotiation Strategy Design	121
5.4	Matching Strategy Algorithm	125
5.5	Experimental Analysis	127
5.6	Conclusions	130
6	Qualitative One-to-Many Multi-Issue Negotiation: Approximating the QVA	133
6.1	Introduction	134
6.2	Definitions	136
6.3	The Qualitative Vickrey Auction	137
6.4	Negotiation Protocols	139
6.4.1	A protocol based on two negotiation rounds	140
6.4.2	Multiple negotiation rounds with multiple sellers	142
6.4.3	A variant without making intermediate agreements public	143
6.5	Experimental Evaluation	145
6.5.1	Experimental Design	145
6.5.2	First Negotiation Protocol	148
6.5.3	Second Negotiation Protocol	149
6.5.4	Third Negotiation Protocol	151
6.5.5	Negotiation Strategy	154
6.6	Related work	156
6.7	Conclusion	159
A	Instantiations of the negotiation protocols with a negotiation strategy	163
7	Eliminating Issue Dependencies in Negotiation Domains	167
7.1	Introduction	167
7.2	Related Work	170
7.3	Utility of Interdependent Issues	171
7.4	Weighted Approximation Method	174
7.4.1	Estimate an Expected Outcome	176
7.4.2	Select Weighting Function	178
7.4.3	Compute Approximation of Utility Space	180
7.4.4	Analyze Difference Δ with Original Utility Space	182
7.4.5	Bid Search Algorithm	183
7.5	Experimental Evaluation	185
7.5.1	Case Study	185
7.5.2	Impact on Outcome Deviation	187

7.5.3	Impact on Computational Cost	190
7.5.4	Tunning Coumputational Costs	191
7.6	Conlusions	192
8	Conclusions and Future Work	197
	Samenvatting	203
	Acknowledgements	209

Chapter 1

Introduction

1.1 Overview

The central aim of this thesis is the design of generic and efficient automated strategies for two-party negotiations in which negotiating parties do not reveal their preferences explicitly. A strategy for negotiation is the decision mechanism for determining the actions of a negotiator. Generic refers to the idea that the strategy needs no forehand knowledge about the opponent or the domain of negotiation. A strategy thus should be generic in the sense that it can be successfully applied to any negotiation domain and fine-tuned to domain-specific features to produce even better results. Efficiency refers to the fact that the strategy should be able to negotiate effectively against another automated agent or human negotiator and obtain an outcome that cannot be improved for both parties. The design of the negotiating strategy that is proposed in this thesis is based on analyses of the state-of-the-art negotiation strategies using an analytical method that is also proposed in this work. The method significantly extends existing negotiation benchmarks by analysing dynamic properties of a negotiation strategy. One of the main findings of the analysis, in line with the management and social science literature on negotiation [20, 23], is that the strategy should learn the opponent's preferences in order to increase the negotiation efficiency. We applied our results in learning the opponents' profiles in a one-to-many negotiation setting. We additionally addressed the problem of issue-dependencies. Issue dependencies form an insurmountable barrier for the state of the art negotiation strategies [9]. Therefore, we developed an approximation method to eliminate dependencies. This part of the research seems a side track, however it was fundamental that we address this problem to prove the scalability and applicability of our research results.

In summary, the research questions underlying this thesis are the following:

1. How can we design state of the art automated strategies for multi-issue two-party negotiations in which only bids are exchanged?
2. What analytical framework is essential for the develop of such automated strate-

gies?

3. Can we learn the preference profile of the opponent given only a sequence of bids exchanged?
4. Can we effectively use preference profiles of the opponent in automated bidding strategies?
5. Can we extend our results to one-to-many negotiation?
6. Can we find a way to approximate negotiation spaces with issue dependencies by spaces without such dependencies?

Negotiation is a type of interaction between two or more self-interested agents (each with its own aims, needs, and viewpoints) seeking to discover a common ground and reach an agreement to settle a matter of mutual concern or resolve a conflict (cf. [20]). People negotiate in their personal life as well as in their business life [23]. Even though most people regard themselves to be effective at negotiation numerous experiments show that people often “leave money on the table” (cf. [23]). The advances in the areas of Artificial Intelligence [26] and Management Studies [23] inspired the first assistance tools for humans in negotiation, and the development of software agents that negotiate on their behalf.

This thesis focuses on generic and efficient bidding strategies for single sessions bargaining between two negotiators. The bidding strategies can be used by negotiating software agents. The stress on single session negotiations is motivated by the fact that various important negotiations in real life are of the single session type; e.g., buying a house or a car, negotiating for a job. From a technical point of view the restriction to single session bargaining implies that we cannot learn from previous experiences with the same opponent¹.

In this thesis we argue that the development of generic efficient bidding strategies requires an analytic framework for thorough evaluation of bidding strategies. For this purpose we developed the General Environment for Negotiation with Intelligent multi-purpose Usage Simulation (GENIUS). We show that a proper analysis of negotiation strategies includes the dynamics of the negotiation instead of only studying the outcomes of negotiations as is typically done in the state of the art of automated negotiation. For this purpose we developed a range of dynamic properties that have proved their usefulness in our analyses and which are included in the analytical environment that is part of the GENIUS framework.

Our analysis of the state of the art in automated two-party bargaining strategies revealed the following important criteria for developing generic and efficient bidding strategies:

- Knowledge about the opponent is essential to reach near Pareto-efficient outcomes.

¹For ease of understanding we call the two parties in two-party negotiations the user and the opponent. The software agents we develop always act on behalf of the user. The other negotiating party is called the opponent. We do this in full understanding of the reasons of the Harvard Business School of avoiding the term opponent.

- The strategy must be tested in domains with different characteristics. For this we developed domain characteristics such as: predictability, size, and interdependencies. Furthermore, we included in GENIUS a repository of domains.
- The strategy must be tested against a range of strategies, as well as against humans. Therefore, we included in GENIUS a repository of strategies.
- The strategy must be tested for a range of profiles (for user and opponent). Therefore, we included in GENIUS a repository of profiles per domain.

As many of the state of the art negotiation strategies are incapable of handling interdependent negotiation issues, we developed an approximation method for eliminating issue-dependencies. This method can be combined with those strategies that search for bids of a particular utility before making a bid to the opponent.

Our analysis showed that state of the art bargaining strategies are in essence concession-based strategies. This means that these strategies typically do not respond to the opponent's negotiation moves to signal how acceptable these moves are. These strategies do not always ensure that a concession is made only if the other party has similarly made concessions. Moreover, the developers of state of the art negotiation strategies do not focus on the fact that the chance of an agreement should be maximized in order to reach an acceptable agreement in the negotiation. In this thesis we developed and tested a generic and behaviour-based strategy that explicitly takes these concerns into account. The strategy is called the Nice Matching Strategy (NMS) as it uses a tit-for-tat approach to safeguard a good outcome for the agent itself but also makes so-called nice moves to maximize the chance that a proposal is accepted by the opponent. The NMS strategy uses a technique based on Bayesian learning called Bayesian learning algorithm for Opponent Preferences (BOP) to learn the preferences of the opponent. NMS uses this constructed opponent model to implement a kind of mirroring-strategy. The mirroring idea is an advanced variation of a Tit-for-Tat strategy. Our analysis shows NMS to be superior to the state of the art.

Learning opponent preferences during a negotiation, i.e., while bids are exchanged, is essential in the context of single session negotiations. This is a finding of our analysis method for negotiation strategies. Our analysis also shows that without knowledge of the opponent model generic bidding strategies cannot be efficient. This learning goal is particularly challenging as we assume that negotiations are closed, i.e., the only information available for this learning process consists of the bids exchanged. BOP has been evaluated both in a two-party negotiation as well as in a one-to-many negotiation setting. The BOP learning mechanism that we developed has been evaluated using the GENIUS environment and has been shown to provide good approximations of the opponent preferences during single sessions.

The method we used to develop generic and efficient bidding strategies is based on an iterative approach in which GENIUS plays a central role:

1. First, by applying GENIUS we identify strengths and weaknesses (inefficiencies) of

existing bidding strategies.

2. Second, the analytical results thus obtained are used to identify new techniques that increase the efficiency and are used to develop improved strategies.
3. Third, the development kit of GENIUS is used to implement the techniques proposed.
4. Fourth, the validity of the new strategies is tested and analyzed by using it to negotiate against the strategies and in various negotiation domains collected in GENIUS repositories by running a GENIUS tournament.

We end this summary by listing the key features of GENIUS. The GENIUS environment has been developed during the course of this PhD project as a full fledged research tool. Its main features are:

- It supports the implementation of new strategies, which facilitates and improves the speed of testing new strategies in the GENIUS environment.
- It includes repositories of negotiation domains, preference profiles per domain, and strategies that facilitate analyzing negotiating agents in a range of different setups.
- It provides graphical user interfaces for constructing and adding new negotiation domains and preference profiles.
- It provides graphical user interfaces which allow humans to negotiate against known or unknown opponents, where opponents can be either humans or software agents.
- It provides a tournament environment that allows a user to manage running various strategies against each other in combination with any selection of domains and preference profiles.
- It provides an analytical environment that assists the researcher in analysing the data obtained from running an environment.

The GENIUS environment, moreover, is used in the education of Master students of the TU Delft and Bar-Ilan University, Israel.

The rest of the chapter provides the necessary background information on negotiation in general as well as the basic concepts frequently used in the thesis. Some aspects are treated in more details as they form the core of our research: knowledge about the domain of negotiation and the opponent, and negotiation strategies and tactics. Finally, this chapter provides an overview of the chapters in the thesis as well as a list of publications underlying the chapters.

1.2 Negotiation: the basics

A negotiation takes place in a certain *domain*, e.g., real-estate, job negotiation, etc. A *negotiation domain* is a detailed description of a conflict to be resolved by a negotiation. Typically, a negotiation domain is represented by a set of negotiation issues. A *negotiation issue* is a topic of discussion that is of a particular interest in a negotiation, e.g., price, quantity, delivery date, etc. Each issue has a range of alternatives or options, one of which must ultimately be agreed upon by the negotiating parties in order to achieve a *compromise*, that is an agreement reached by mutual concessions.

The number of issues varies from one domain to the next. The number of issues, the range of values per issue, and the possible interdependencies between issues determine the complexity of a conflict of interests. Negotiations can be split into two classes with respect to the number of issues: *single-issue* and *multi-issue* (i.e., more than one issue) negotiations. Single-issue negotiation is also known as the “splitting a pie” game [17] and is concerned with the division of a single good, e.g. money. Such negotiations Raiffa calls “win-lose” negotiations [20] due to the fact that increasing a share of a pie for one party means decrease of a share for the other (often called opponent in this thesis). Negotiations in domains with multiple issues can be “win-win” negotiations meaning that by trading less important issues for more important issues both parties benefit from the agreement. An analysis of such negotiations is more complex than that of the single-issue case due to the exponential increase in possible outcomes [20]. One aspect of the art of negotiation is the ability to see more issues than are initially obvious. A famous example is about fruit [14]. Suppose two people have to share a kilo of cherries. The obvious issue to negotiate about is how many grams of cherries each will get. Thus a typical example of a win-lose negotiation: every cherry that one gets is lost to the opponent. However, a smart negotiator will see if the one issue can be split in two and might suggest to split the issue into fruit flesh versus kernel. In case that negotiator is only interested in the kernels (to create bed-heaters), whereas the opponent is only interested in the fruit flesh to make cherry pies it is easy to see that a perfect win-win negotiation is taking place. Of course, life is not this easy in general. However, in general the more issues, the more opportunities there are for win-win solutions.

A full and automated analysis of a negotiation requires a formal representation of the domain (i.e., issues, ranges per issue), and the preferences of the negotiators. A number of preference representation formalisms have been proposed (cf. [2, 13, 19, 24]). These representations can be divided in two classes: qualitative and quantitative models. *Qualitative models* typically deal with ordinal ranks [2], e.g., they can be used to express relations between agreements such as: “agreement a is preferred over agreement b , which in its turn is preferred over agreement c ” ($a < b < c$). Recently, qualitative models have been developed for more complex relations between alternatives. CP-nets for example, [3] allow the expression of dependencies between alternatives of different issues.

This thesis is concerned with quantitative models of preferences. Such *quantitative models* are mathematical functions that map negotiation outcomes to a numerical scale, see

e.g., [24]. Various types and structures of utility functions have been proposed [13]. The type of functions that belongs to the class of the *linear additive functions* is most commonly used and also used throughout this thesis (except in the Chapter on issue dependencies). An advantage of this type of function is that there are computationally efficient algorithms for searching a bid with a particular utility value when preferences are represented by these functions. Linear additive functions represent relatively simple preferences in which the contribution of every issue to the utility is linear and does not depend on the values of other issues (no issue dependencies). Such utility functions are widely and successfully used in the negotiation and decision making literature [13, 21]. Quantitative utility functions also are the dominant choice for preference models in automated negotiation as we found through our extensive literature study, for this we refer to all references on negotiation in this thesis. Another advantage of linear additive utility functions is that effective preference elicitation techniques exist for these utility functions, see [13, 19]. For this reason, utility functions have been selected as the preference representation model in this thesis.

Note that preference elicitation is a field of research of its own, see e.g., [13], and is not addressed in this thesis. In this work we assume that the issues to be negotiated have been established and that preference profiles over these issues are given and adequately represent the preferences of the negotiation parties.

In this thesis we distinguish three types of negotiations: human, automated and mixed negotiations. By *human negotiations* we mean a setup in which humans resolve a conflict without intervention from others. In an *automated negotiation* the negotiating parties are represented by automated software *agents* [26] that negotiate on behalf of their parties. In a *mixed negotiation* one party is human, the other is a software agent.

The interaction between negotiating parties is regulated by a *negotiation protocol* that defines the rules of encounter and dictates when and what information can be exchanged. Every party must accept and agree to conform to these rules. In this thesis we use the *alternating-offers protocol* for bilateral negotiation as proposed in [17]. In the alternating-offers protocol the negotiating parties exchange offers in turns. Every turn a negotiating party has three options: (i) accept the last opponent's offer, (ii) respond with a counter-offer, or (iii) stop the negotiation.

The number of negotiating parties is used to classify negotiation settings. In this thesis we predominantly address the *one-to-one* or *bilateral* negotiations involving two parties. However, in Chapter 6 we show how some of the techniques proposed in this thesis can be used in one-to-many negotiations where a buyer faces several suppliers and needs to select one.

1.3 Research approach

Two major approaches to the design of negotiation strategies have been proposed in the negotiation literature:

1. Game-theoretic;
2. Heuristic methods.

In this thesis we combine these two. We study the addressed problems and proposed solutions using an analytical, mathematical approach. However, the negotiations that we deal with in this thesis are often too complex to be solved using a purely analytical approach. In such cases, experimental validation is used. In the design of experimental setups we ensure sufficient coverage of the considered possible negotiation settings. Computational complexity can be due to issue dependencies or due to the inherent computational complexity of some algorithms. We use heuristic methods to address both problems.

The *game-theoretic* approach to the negotiation problem is to study the rational decisions an agent can make. The overall negotiation outcome depends critically on the choices made by all negotiating parties. This implies that in order for an agent to make the choice that optimises its outcome, it must take into account the decisions that other agents may make, and must assume that they will act so as to optimise their own outcome. This means that the agent has to take into account the private valuations that opponents have of the negotiation issues, their private deadlines for making a deal, and so on. Game theory gives us a way of formalising and analysing such models.

Single-issue negotiations are addressed e.g., by [17, 19, 20]. Following the game theoretic approach, according to which single-issue negotiations are zero-sum games, equilibrium strategies have been proposed [17]. A multi-issues negotiation is not a zero-sum game and is by nature more complex than single-issue negotiations. Furthermore, no dominant strategies have yet been found for multi-issue negotiation [11, 17, 20].

Game theory models assume perfect rationality and complete information [17]. Perfect rationality assumes that if the optimal (rational) solution for the problem is known, that solution is adopted by all rational parties. Sometimes the optimal solution, although it exists, is not known, due to computational costs. Furthermore, if an optimal solution theoretically can be determined, it is only possible under the circumstances that all information is fully known by the agents. This assumption is rarely true in real world cases; agents typically know their own information space, but they do not know that of their opponent. Therefore, the notion of perfect rationality and complete information, although useful in designing, predicting and analyzing properties of a negotiation strategy, is not altogether useful in practice.

The major means of overcoming the aforementioned limitations of game theoretic models is to use *heuristic methods*. Heuristics are methods that find a solution that is as good as possible given the limited time and information available and given the uncertainty of the available information. The methods themselves may either be computational approx-

imations of game theoretic techniques or they may be computational realisations of more informal negotiation models. Examples of such models can be found in [5, 15]. The key advantages of the heuristic approach can be stated as follows:

- the models are based on realistic assumptions; hence they provide a more suitable basis for automation and they can, therefore, be used in a wider variety of application domains;
- the designers of agents, can select an alternative heuristic that is more suitable for a specific negotiation domain and setup.

While heuristic methods do indeed overcome some of the shortcomings of game theoretic models, they also have a number of comparative disadvantages:

- the models might select outcomes that are sub-optimal; this is because they adopt an approximate notion of rationality and because they do not examine the full space of possible outcomes;
- the models need extensive evaluation, typically through simulations and empirical analysis, since it is impossible to predict precisely how an opponent will behave in a wide variety of circumstances.

In our work we use the heuristic instead of the game-theoretic approach to design negotiation strategies, because of the strong limitations imposed by the assumptions made in game-theoretic models. However, we use game-theoretic models (e.g., optimality criteria) to formulate the requirements for efficient negotiation strategies, for the analysis of their performance, and for the design of our research tools.

1.4 Negotiation: outcome analysis

Given a negotiation domain and preference profiles of the negotiators one can formally analyse the outcome of a negotiation. A number of criteria have been proposed in literature (see e.g., [20]).

Figure 1.1 shows a classic graphical analysis of a bilateral negotiation problem that is used throughout this thesis ². The axis of the chart represent utilities of the negotiating parties (the more the better). The solid points refer to possible *negotiation outcomes*. Every negotiation outcome in the negotiation space is a unique combination of values of the negotiation issues. Given the values of the issues a utility value can be calculated per negotiator using a corresponding utility function. The various utility values of a negotiation outcome determine coordinates of the outcome.

To see how well one has performed in a negotiation a number of optimality criteria have been proposed in the literature, cf. [20]. One of the most important criteria is *Pareto*

²The game-theoretic concepts used for bilateral negotiations can be easily extended for the one-to-many and many-to-many cases

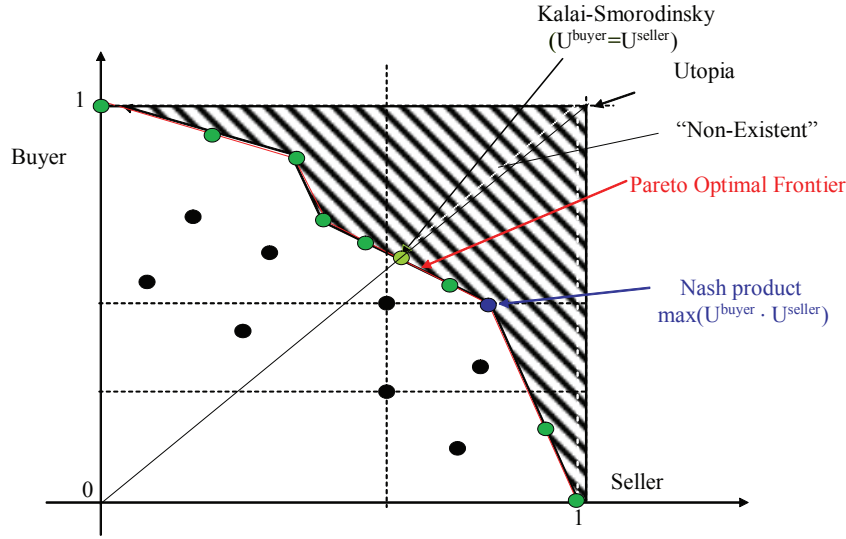


Figure 1.1: A Negotiation Problem and Optimal Solutions

efficiency. A bid is Pareto efficient if given a set of alternatives, no movement from the bid to an alternative exists that can make at least one individual better off without making any other individual worse off. Typically, there exist multiple Pareto optimal solutions that form a Pareto optimal frontier ranging from an outcome in which one negotiator gets everything and the other one gets nothing to an outcome in which the other gets everything and the first gets nothing (see 1.1). To select a single outcome on the Pareto optimal frontier several criteria have been proposed including the Nash product and Kalai-Smorodinsky, see [19] for details.

1.5 Negotiation: knowledge counts

Typically, when multiple issues are involved negotiators assign different priorities to them. For example, in a real-estate domain for a buyer a delivery date of a house can be of a higher importance than a price (within limits). On the other, hand a seller might assign a higher priority to the price issue because she is flexible with respect to timing the sale. Such a difference in issue priorities gives an opportunity of a trade-off between issues. A *trade-off* is a negotiation move that involves mutual losing on less important negotiation issues in return for a gain on issues that have a higher importance. An agreement that is reached by means of trade-offs between issues is called a “win-win” agreement.

Note, that a trade-off is not possible in single-issue negotiations. Some people try to argue that in a real-life there is usually only one issue to negotiate about: an issue of money or a monetary equivalent. According to Thompson, however, “it is a grave mistake to focus on a single issue in a negotiation because, in reality, more issues are at stake in

most negotiation situations” [23]. Indeed, many negotiation experts (cf. [19, 20]) teach negotiators to explore a negotiation domain and come up with new negotiation issues to allow for trade-offs. Negotiators are advised to collaborate in the exploration process, as it is in both their interests for common good. The rule of thumb is “The more issues the more possibilities for various trade-offs between issues”.

Availability of information about the opponent plays an important role. In real-life, usually a negotiator has *incomplete or no information* about the opponent’s preferences. Several reasons exist. First of all, the explicit elicitation of one’s preferences may be difficult [1]; new issues might come up and preferences might change during the negotiation. Modelling preferences is a complex problem [22]. The players may not want to reveal their preferences out of fear of exploitation, now or in future encounters.

Under incomplete information it is impossible to guarantee Pareto efficiency of offers made by a negotiator. As we explained earlier, Pareto efficiency is an important property of offers and a final agreement. Furthermore, negotiation moves made by an opponent cannot be judged by a negotiator objectively without information about the opponent’s preferences. For example, a concession move, that is understood as a sacrifice by one negotiator for a benefit of the opponent, can be in fact a “lose-lose” move for both parties. Such a move can be perceived as unwillingness to concede by the opponent and result in a misunderstanding. We conclude that incomplete information about opponent’s preferences is a major problem for negotiators.

Concluding, the more one knows about the opponent the better the chances of reaching an agreement and the better quality of the agreement that is reached. To some extent the same holds for the domain. For the development of our negotiation strategies we have a need for formalised models of the opponent preferences and of domains. We chose to call those models *Opponent model*, respectively *Domain model*.

In automated negotiations several proposals have been made to overcome the typical incompleteness of knowledge when starting a negotiation. The effectiveness of providing knowledge about the domain of negotiation has been demonstrated in the Trade-off strategy introduced in [6]. In particular, this paper shows that domain knowledge (coded as so-called similarity functions) can be used to select bids that are close to an opponent’s bids, thus increasing the likelihood of acceptance of a proposed bid by that opponent. In this approach, the knowledge represented by similarity functions is assumed to be public. As is to be expected, if similarity functions can be found, the Trade-off strategy outperforms negotiation strategies lacking all information about the domain or the opponent (see Chapter 2). Incorporating public domain knowledge into a strategy, however, still does not take into account the private preferences or priorities that an opponent associates with negotiated issues.

To overcome the limitations of the domain knowledge approach a number of learning techniques have been proposed for the negotiation domain. In [15] a set of “typical” or common preference profiles for the domain at hand is made available to the software agent. The agent uses Bayesian learning to determine which of these common profiles

probably fits the current opponent best. This is a good approach that will work in practice, if such a set of typical opponent profiles is available for the domain at hand.

Another approach to model the opponent's preferences is based on the assumption that some historical data is available which can be used to model an opponent's behaviour. For example, in [16] authors extend the original negotiation strategy of [15] with a learning of common opponent's profiles from historical negotiations. Historical data is also used in [4] to estimate the probability of acceptance of an offer by an opponent using information about accepted and rejected offers by the opponent in the previous negotiations. Unfortunately, such historical information is not available in single session negotiations. In general, historical data is usually collected for a specific negotiation domain and cannot be reused in other domains.

1.6 Negotiation Strategies and Tactics

In addition to knowing about the domain and the opponent's preferences, a good negotiator pays attention to the negotiation tactic of the opponent. Understanding the opponent's tactic helps in finding a fair outcome, but also in finding and taking advantage of the weaknesses of the opponent's tactic. An interesting research topic would be to learn the opponent's tactic. However, we left that for future research.

In this thesis we address two dimensions of bidding tactics, i.e., the concession dimension that varies over conceding as a point of departure to not conceding, and the dimension of using an opponent model (or not). What we leave for future research are the end-of-game problems and acceptance criteria. The aspect of end-of-game problems is related to Time-Dependent-Tactics (TDT) [5].

In this thesis for the concession dimension of bidding tactics we consider concession-based tactics and behaviour-based tactics. By concession-based tactics we mean tactics in which time and again the agent will propose bids that are weakly monotonous concessions with respect to the utility function of the bidder. Simple concession-based bidding strategies concede with every bid they make (e.g., ABMP [12]), more advanced strategies concede when there is no longer an option of improving the bid from the perceived perspective of the opponent without conceding according to the agent own utility function (e.g., Trade-Off [6]).³

A behaviour-based bidding tactic takes the behaviour of the opponent into account. A simple example of such a tactic is the well-known Tit-for-Tat [5]. The QO strategy of [16] uses offers of an opponent to estimate opponent's preferences and then uses this information in a minimax fashion when generating next offers.

³A complicating factor in such strategies is that generally those strategies don't use an opponent model. As a result, in a multi-issues negotiation domain a concession with respect to the utility function of one party does not necessarily mean a concession with respect to the utility function of the opponent ([10]). A domain model is not enough for a strategy to be able to recognize such situations.

The second dimension distinguishes between strategies that make use of an opponent model and those that do not. The ABMP strategy belongs to the type of strategies that do not model an opponent. The Trade-Off strategy assumes that its domain model accurately models a part of the opponent's preferences. The strategy uses similarity measures of values of issues derived from the domain knowledge. It maximizes the opponent's utility by maximizing similarity of the own and the opponent's bids. The QO strategy uses a set of possible opponent preference profiles to model an opponent. The strategy uses Bayesian learning to estimate probabilities of correctness of the profiles in the set. A profile with the highest probability is used to generate the next offer.

When analysing the state of the art in negotiation strategies for automated negotiations, we found that the underlying tactic for each of them is concession-based, with the exception of the QO strategy. Our results in opponent modelling open the possibility of studying behaviour-based tactics, see Chapter 5.

1.7 Contribution of the Research

The central problem addressed in this thesis is the design and engineering of generic and efficient negotiation strategies for automated negotiations. The main contributions of this thesis are:

- a method for the analysis of outcomes and dynamics of multi-issue negotiations and negotiation strategies;
- criteria for developing generic efficient negotiation strategies based on an analysis of the existing strategies
- General Environment for Negotiation with Intelligent multi-purpose Usage Simulation(GENIUS)
- a learning technique, called BOP, that learns opponent's preferences in single-session negotiations
- an efficient and behaviour-based negotiation strategy, called NMS, that makes use of the learning technique BOP for opponent's preferences in a single-session negotiation
- the application of the learning technique to one-to-many negotiations
- an approximation method for non-linear utility spaces with interdependent issues

1.8 Thesis Overview

This thesis is based on a collection of articles. Except for the layout of the papers, they are left unchanged. Every chapter covers a specific aspect of this thesis and can be read

independently of each other. Therefore, there is some overlap in introductions and definition of the used concepts. Authors of the articles are listed in alphabetical order and their contribution is deemed of equal value.

We start with a proposal of an analytical method and a research tool. In Chapter 2, we describe the analytical method, apply it to the state-of-the-art negotiation strategy, and formulate design guidelines for efficient negotiation strategies. In Chapter 3 we introduce an architecture for a negotiation research framework to assist the design of the strategy. Next, we propose the efficient negotiation strategy. Chapter 4 proposes a learning technique for the opponent's preferences. In Chapter 5 the learning technique is combined with a bidding strategy to increase robustness of the strategy. In Chapter 6 we apply the learning technique in one-to-many negotiation setups, in particular the Qualitative Vickrey Auction. Finally, Chapter 7 proposes an approximation technique that can be used to adapt algorithms to non-linear utility functions.

1.8.1 An Analytic Framework of Negotiation Dynamics and Strategies

In Chapter 2 we introduce the method for the analysis of dynamic features of negotiating strategies. By looking into the dynamics of strategies the method extends the existing analytical approaches of negotiation outcomes (see e.g., [20]) by helping us to understand why these outcomes are obtained. The method is based on a classification of negotiation moves and provides a number of useful metrics. These metrics in turn are used to define more complex dynamic properties of the strategies under evaluation.

According to [11], there is a need for the development of a best practice repository for negotiation techniques. That is, a coherent resource that helps to determine which negotiation techniques are best suited for a given type of problem or domain (much like the way the TREC data sets and conference works [25]). Indeed, in Chapter 2 we prove that a negotiation strategy designed using the heuristic-based approach has to be tested in different negotiation settings, i.e., on a wide range of negotiation domains and against various opponent's negotiation strategies. Therefore, in the method we identify important negotiation factors that influence negotiation behaviour, including: size of the negotiation domain, predictability of the preferences, opposition of the preferences, opponent's strategy. These factors play an important role in this work and will be used throughout this thesis.

The method is applied to the state-of-the-art negotiation strategies. From the analysis, we conclude that each analyzed strategy has its strengths and weaknesses that depend on the negotiation domain and/or opponent's strategy. We further conclude that it is impossible to avoid unfortunate moves (i.e., bids that make things worse for both parties) without sufficient domain knowledge or a model of the negotiation partner. We use this analysis to formulate criteria for an efficient generic negotiation strategy. The method is applied to the Bayesian Smart strategy that is based on the learning technique proposed in Chapter

4.

1.8.2 An Open Negotiation Architecture for Heterogeneous Agents and A Negotiation Testbed

In Chapter 3 we propose an open negotiation architecture for heterogeneous agents. The architecture allows easy development and integration of existing negotiating agents using *design patterns*. The architecture is implemented in GENIUS, a General Environment for Negotiation with Intelligent multi-purpose Usage Simulation.⁴ The core functionality of the system includes:

- specification of negotiation domains and preference profiles;
- simulation of bilateral negotiations between agents;
- analysis of the negotiation outcomes and negotiation dynamics;
- human-computer negotiations;
- human-human negotiations via internet

GENIUS proved its worth in education at the Radboud University Nijmegen (The Netherlands), the Bar-Ilan University (Israel), and the Delft University of Technology (The Netherlands).

An analytical toolbox integrated in GENIUS can be used for graphical analysis of a negotiation outcome and calculates optimal solutions, such as the Pareto efficient frontier, Nash product and others. The toolbox may be used to visualize a negotiation process and creates an extensive log. In addition to the outcome analysis, the toolbox implements the metrics of negotiation dynamics as described in Chapter 2.

Unlike the methodology proposed in [18] where an agent is designed for a specific negotiation setting we propose an architecture to support the design of generic negotiation agents. For that purpose, GENIUS has a repository of negotiation domains, preference profiles and negotiation strategies that can be used to organize tournaments. Several negotiation domains are currently collected in the repository of GENIUS. Each domain has at least two preference profiles required for bilateral negotiations. The number of issues in the domains ranges from 3 to 10, where the largest negotiation domain in the repository is the AMPO vs City taken from [19], and has over 7,000,000 possible agreements. The repository of strategies currently contains six automated negotiation strategies, such as the ABMP strategy [12], the Zero-Intelligence strategy [7], the QO-strategy [15], the NMS strategy proposed in Chapter 5. The repositories of domains and of agents allow agent designers to test their agents on various domains and against various agents and humans.

⁴Previous versions of the system were known under the name SAMIN (System for Analysis of Multi-Issue Negotiation). The most recent version of GENIUS can be downloaded from <http://mmi.tudelft.nl/negotiation>.

1.8.3 Learning Opponent's Preferences in Negotiation

In Chapter 2 we concluded that one of the main necessities of an efficient strategy for closed negotiation is a model of the opponent's preferences. In addition, we identified a number of requirements for efficient negotiation strategies. In Chapter 4 we propose an effective Bayesian learning algorithm for Opponent Preferences (BOP) in a closed single-session negotiation. The learning technique we designed can be integrated in any negotiation strategy to increase the efficiency of its offers. In other words, BOP can be used to generate an offer that has a certain (approximated) utility for the opponent. In particular, such strategies as Trade-Off can be made more efficient by locating offers on the Pareto optimal frontier (approximately).

For feasibility we had to make some design choices to assume that opponent's preferences models have a certain structure and that the opponent will use a concession tactic. The assumption on structure reduces the hypothesis space of possible utility functions. The simplifying assumption we made is that opponent's preferences can be modelled using linear additive utility functions in which each issue is evaluated using one of the following three types of evaluation functions: *downhill shape* (minimal issue values are preferred over other issue values, e.g., of price and delivery time for a buying agent), *uphill shape* (opposite to the downhill shape), *triangular shape* (a specific issue value somewhere in the issue range is valued most and issues to the left and right are valued less).

Our simplifying assumption is that an opponent follows some kind of concession-based strategy. Although this assumption may not be realistic for every counteroffer they make, negotiators do have to make at least some concession steps in order to reach an agreement. Moreover, in game-theoretic and heuristic-based approaches to negotiation it is commonly assumed that negotiating parties use a concession-based strategy [5, 17]. The learning algorithm allows for the incorporation of prior available opponent knowledge but does not require any such knowledge.

To show the effectiveness of our approach to learn an opponent model and that it can be used to find a good counteroffer the BOP learning technique was combined with an existing strategy [5] and integrated in a negotiating agent, called the Bayesian Smart agent. The results of using Bayesian Smart in a negotiation setting show the effectiveness of using an opponent model in a negotiation strategy to improve the efficiency of the bidding process.

Although we showed the effectiveness of BOP in combination with negotiation strategies, including BOP does not address all weaknesses of such strategies. For example, the Bayesian Smart strategy can be easily exploited by an opponent by submitting the best offer for itself as the last offer before the negotiation deadline. Thus, a more robust negotiation strategy is needed.

1.8.4 Negotiation Strategy

Using the efficient learning technique for opponent's preference, in Chapter 5 we propose a robust technique to determine the size and direction of the negotiation moves. This technique allows us to formulate a negotiation strategy that responds to offers of an opponent in a behaviour-based way. Three criteria for the designed strategy are formulated.

Firstly, the strategy should be *efficient*. As before, here efficiency means that the strategy should always propose offers on the Pareto frontier that are approximated using a model of opponent's preferences. Otherwise, both parties would leave money on the table. Therefore, for each negotiation move the strategy moves to the approximated Pareto frontier. Secondly, moves made by the strategy should be *transparent* to the opponents. The strategy achieves this by a simple response mechanism that mirrors an opponent's move. Transparency can be achieved by using available knowledge about the preference profile of the opponent. Finally, the strategy should *maximize the chance of an agreement and should avoid exploitation*. To maximize the chance of an agreement the strategy always moves to the approximated Pareto frontier, that maximizes the utility of the opponent relative to a particular utility for the agent itself.

Obviously, the strategy requires information about the opponent's preferences. For that purpose, the BOP learning algorithm proposed in Chapter 4 is used. Of course the strategy can be combined with alternative, future opponent modelling techniques. The effectiveness of the strategy has been validated experimentally in our usual tournament setup, using domains of different characteristics and a number of different negotiation strategies. The results show that the strategy is able to realize significant increases in utility.

1.8.5 Approximating the Qualitative Vickrey Auction by a Negotiation Protocol

In Chapter 6, we apply the BOP learning algorithm to a one-to-many multi-issue negotiation setting. We look at a particular instance of this more general problem and study a particular auction mechanism called a Qualitative Vickrey Auction (QVA) [8]. This auction is a generalization of the well-known Vickrey auction to a general complex multi-issue setting where payments are not essential. This also means that "pricing out" is not an option to elicit preferences. This implies that in case of somewhat cooperative negotiation domains a buyer and a seller(s) have the opportunity of "win-win" outcomes. The QVA requires the buyer to publicly announce its preferences and in that case it can be shown that the outcome is efficient and the mechanism is strategy-proof. However, sometimes it is too costly, impossible, or undesirable to publicly announce such preferences. In Chapter 6 we study various multi-bilateral negotiation mechanisms and show how the BOP learning algorithm can be used to drop this requirement.

The main idea is that the (efficient) outcome of the QVA may be approximated by a negotiation protocol that consists of multiple negotiation rounds in which sellers are provided

an opportunity to outbid the winner of the previous round. The main assumption that we need to make to obtain this result is that the negotiating agents are able to (privately) learn part of the preferences of their opponents during a negotiation session. We show experimentally that each of these mechanisms is able to approximate the efficient outcome as defined by the QVA. Additionally, experiments are performed that show that a negotiating agent that uses learning significantly outperforms a Zero Intelligence strategy [7].

1.8.6 Eliminating Issue Dependencies in Negotiation Domains

In Chapters 2 - 6 we focused on negotiation domains for which preferences can be modelled by means of linear utility functions, i.e., utility of alternatives of a single issue does not depend on any other issue. In some negotiation domains preferences can have a more complex structure.

In Chapter 7 we address a problem of negotiation in domains with interdependent issues. Interdependencies between issues in a negotiation domain result in non-linear utility spaces. Finding good bids in such spaces is a computationally complex problem which grows exponentially with the number of issues.

To allow existing negotiation strategies designed for negotiation domains with independent issue to deal with utility spaces with issue dependencies we propose an off-line approximation method that approximates the original non-linear utility space with a corresponding linear space. The approximation method minimizes distance between the original space and its linear approximation. It is based on the following observations. First, not all bids are equally important for negotiation: there are some bids which are not acceptable for the agent and some that are too optimistic to be an outcome of the negotiation. In effect, it is possible to indicate an expected region of utility of the outcome. Second, we conjecture that real life cases have a structure that is far from random that can be modelled by utility functions mostly linear additive functions, with a few issues dependencies.

This chapter proposes a bid search algorithm based on the weighted averaging approximation method. Using an approximation, however, always comes with a risk that a bid is proposed (and accepted by the other party) that seems to have a good utility, but in fact, in the original utility space has a much lower utility. A checking procedure is introduced into the bid search algorithm that offers a way to avoid this risk at the cost of additional computations. The parameters of the checking procedure allow the tuning of the negotiation algorithm to increase either the computational efficiency or decrease the risk of erroneous bids. Derived from experimental results, we propose specific values for these parameters that ensure a balance between computational costs and outcome deviation (in terms of utility) in many domains. Finally, we present experimental results that show that the approach of adding a checking procedure to the negotiation algorithm is scalable and allows an agent to negotiate about high-dimensional utility spaces with issue dependencies.

1.9 Publications related to each chapter

All the Chapters of this thesis are based on publications in scientific journals or refereed book chapters.

Chapter 2: Koen Hindriks, Catholijn Jonker and Dmytro Tykhonov, “Negotiation Dynamics: Analysis, Concession Tactics, and Outcomes”, In: *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT’07*, pp. 427-433, 2007.

Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov, “Analysis of Negotiation Dynamics”, In: *Proceeding of the Eleventh International Workshop on Cooperative Information Agents (CIA’07)*, ISBN 3-540-75118-1, LNAI, no. 4676, pp. 27-35, Springer, 2007.

Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov, “Let’s Dance! An Analytic Framework of Negotiation Dynamics and Strategies”, submitted to: *Web Intelligence and Agent Systems - An International Journal*.

Chapter 3: Koen Hindriks, Catholijn Jonker, Dmytro Tykhonov, “Towards an Open Negotiation Architecture for Heterogeneous Agents”, In: *Proceeding of the Twelfth International Workshop on Cooperative Information Agents (CIA’08)*, Springer, 2008.

Raz Lin, Sarit Kraus, Dmytro Tykhonov, Koen Hindriks, Catholijn M. Jonker, “Supporting the Design of General Automated Negotiators”, In: *Proceedings of the Second International Workshop on Agent-based Complex Automated Negotiations (ACAN’09)*, May 2009.

Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, Raz Lin, Dmytro Tykhonov, “GENIUS - Negotiation Environment for Heterogeneous Agents”, In: *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, May 2009.

Koen V. Hindriks, Catholijn M. Jonker and Dmytro Tykhonov, “A Multi-Agent Environment for Negotiation”, In: *Multi-Agent Programming: Languages, Tools and Applications*, pp. 333-363, Springer US, 2009.

Chapter 4: Koen Hindriks and Dmytro Tykhonov, “Opponent Modelling in Automated Multi-Issue Negotiation”, In: *AAMAS’08*, 2008.

Koen Hindriks, Catholijn Jonker, Dmytro Tykhonov, “BOP: an Effective Bayesian Learning Algorithm for Opponent Preferences”, submitted to: *Journal of Artificial Intelligence Research*.

Chapter 5: Koen Hindriks, Catholijn Jonker, Dmytro Tykhonov, “Using Opponent Models for Efficient Negotiation (Extended Abstract)”, In: *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, May 2009.

Koen Hindriks, Catholijn M. Jonker and Dmytro Tykhonov, “The Benefits of Opponent Models in Negotiation”, In: *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, September 2009.

Chapter 6: Koen Hindriks, Dmytro Tykhonov and Mathijs de Weerd, “Approximating an Auction Mechanism by Multi-Issue Negotiation”, In: *Proceedings of the First International Working Conference on Human Factors and Computational Models in Negotiation (HuCom 2008)*, December 2008.

Koen V. Hindriks and Dmytro Tykhonov and Mathijs de Weerd, “Approximating the Qualitative Vickrey Auction by a Negotiation Protocol”, In: *Proceedings of The Eleventh International Workshop on Agent Mediated Electronic Commerce (AMEC 2009)*, May 2009.

Koen V. Hindriks and Dmytro Tykhonov and Mathijs de Weerd, “Approximating the Qualitative Vickrey Auction by a Negotiation”, accepted for *Special Issue of Group Decision and Negotiation Journal*, 2010.

Chapter 7: Hindriks, K.V., Jonker, C.M., and Tykhonov, D., “Eliminating Interdependencies between Issues for Multi-issue Negotiation.”, In: *Proceedings of the Tenth International Workshop on Cooperative Information Agents*, vol. 4149, no. Lecture Notes in Artificial Intelligence., Edinburgh, United Kingdom, September 11-13 2006.

Hindriks, K.V., Jonker, C.M., and Tykhonov, D., “Reducing Complexity of an Agent’s Utility Space for Negotiating Interdependent Issues”, *InterJournal for Complex Systems*, GALetter manuscript-number 1821, 2006.

Koen V. Hindriks, Dmytro Tykhonov, Catholijn M. Jonker, “Avoiding Approximation Errors in Multi-Issue Negotiation with Issue Dependencies”, In: *Proceedings of the First International Workshop on Agent-based Complex Automated Negotiations’08*, Estoril, Portugal, pp.64-79, 2008.

Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov, “Eliminating Issue Dependencies in Complex Negotiation Domains”, accepted to: *Special Issue of Multiagent and Grid Systems - an International Journal*, 2010.

Bibliography

- [1] M. Bichler, J. Lee, H. Lee, and J. Chung. Absolute: An intelligent decision making framework for e-sourcing. 2001. San Jose, CA.
- [2] Craig Boutilier. Toward a logic for qualitative decision theory. In *In Proceedings of the KR'94*, pages 75–86. Morgan Kaufmann, 1992.
- [3] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [4] R. Carbonneau, G.E. Kersten, and R. Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2):1266–1273, February 2008.
- [5] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [6] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
- [7] Dhananjay K. Gode and Shyam Sunder. Allocative efficiency in markets with zero intelligence (zi) traders: Market as a partial substitute for individual rationality. In *Journal of Political Economy*, 100(1):119–137, 1993.
- [8] Paul Harrenstein, Tamas Mahr, and Mathijs M. de Weerd. A qualitative vickrey auction. In Ulle Endriss and Goldberg Paul W, editors, *Proceedings of the 2nd International Workshop on Computational Social Choice*, pages 289–301. University of Liverpool, 2008.
- [9] K Hindriks, C. M. Jonker, and D. Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. In Klusch, M., Rovatsos, M., , Payne, and T.R., editors, *Cooperative Information Agents X, Proceeding of Tenth International Workshop on Cooperative Information Agents*, pages 301 – 316, 2006. Edinburgh, United Kingdom.
- [10] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Analysis of negotiation dynamics. In *Proceedings of 11th International Workshop CIA 2007 on Cooperative Information Agents, Delft, The Netherlands*, volume 4676 of *Lecture Notes in AI*. Springer-Verlag, 2007. Invited paper.
- [11] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10:199–215, 2001.
- [12] Catholijn M. Jonker and Jan Treur. An agent architecture for multi-attribute negotia-

- tion using incomplete preference information. *Autonomous Agents and Multi-Agent Systems Journal*, 15:221–252, 2001.
- [13] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives*. Cambridge University Press, 1976.
 - [14] Deborah M. Kolb. Love for three oranges. *Negotiation Journal*, 1995.
 - [15] Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence Journal*, 172(6-7):823–851, 2008.
 - [16] Raz Lin, Yinon Oshrat, and Sarit Kraus. Investigating the benefits of automated negotiations in enhancing people’s negotiation skills. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, May 2009.
 - [17] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
 - [18] I. Rahwan, L. Sonenburg, N. R. Jennings, and P. McBurney. Stratum: A methodology for designing heuristic agent negotiation strategies. *International Journal of Applied Artificial Intelligence*, 21(6):489–527, 2007.
 - [19] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.
 - [20] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2003.
 - [21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition edition, 2002.
 - [22] J. E. Teich, H. Wallenius, J. Wallenius, and O. R. Koppius. Emerging multiple issue e-auctions. *European Journal of Operational Research*, 159, 2004.
 - [23] L.L. Thompson. *The Heart and Mind of the Negotiator*. Pearson Prentice Hall, 2005.
 - [24] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton, NJ. Princeton University Press, 1944.
 - [25] E. M. Voorhees and Lori P. Buckland, editors. *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*. NST, 2008.
 - [26] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons, second edition, 2009.

Chapter 2

Let's Dans! An Analytic Framework of Negotiation Dynamics and Strategies

Abstract. The “negotiation dance”, as Raiffa calls the dynamic pattern of the bidding, has an important influence on the outcome of the negotiation. The current practice of evaluating a negotiation strategy is to focus on fairness and quality aspects of the agreement. In this article we present the framework DANS (Dynamics Analysis of Negotiation Strategies) for the analysis of the dynamic patterns of the bidding as a means to evaluate the strengths and weaknesses of negotiation strategies for bidding. The method provides the tools to perform a detailed and quantified analysis of a negotiation between two agents in terms of dynamic properties of the negotiation trace. The classification of negotiation steps in the dance plays a central role in the analysis. The method can be applied to tournaments, but can also be used to analyze single 1-on-1 negotiation sessions. The sessions can be played by humans or by software agents. Using DANS we show that some strategies are sensitivity to the bidding behaviour of the opponent, and some depend on a correct model of the opponent. DANS helped us discover that domain characteristics are important for the analysis of strategies. Some strategies rely heavily on some domain assumptions. Furthermore, the results illustrate that having domain knowledge is not always enough to avoid making unintentional steps. The method is demonstrated in the analysis of three strategies from the literature ABMP, Trade-Off and Bayesian Agent.

2.1 Introduction

The negotiation dance of exchanging successive offers by negotiating parties affects the negotiation outcome [15]. To gain more insight in the negotiation dynamics, in [1] a classification of negotiation moves was introduced in order to characterize and compare the bidding process of humans and software agents. The results show an overall similarity of the bidding style of humans and the Agent-Based Market Places (ABMP) strategy, a concession-oriented negotiation strategy, see [11]. However, the analysis did not provide

insights in why the different kinds of moves were made, nor did it help us understand why and to what extent these moves affect the outcome of the negotiation. As far as we know, no analytical methods exist that do provide the desired insights.

The analysis method introduced in this paper is a concrete step towards providing such insights. It extends the work presented in [1] by extending and providing a precise characterization of the negotiation move classification and by providing some useful metrics. These metrics in turn are used to define more complex dynamic properties of the negotiation dance to facilitate the analysis of various dynamic properties of the strategies under evaluation.

Other analytical methods used in the literature typically assess the performance of negotiation strategies in terms of fairness and quality aspects of the agreement (if any) that agents reach. Aspects considered are who wins, the distance of the outcome to the Pareto Efficient Frontier, the Nash Product, and e.g. the Kalai-Smorodinsky Point (see section 2.2.1 for details). Formal definitions of these concepts can be found in e.g., [15]. Such measures of evaluation focus on the negotiation outcome.

Instead, the concepts introduced here are intended to facilitate the analysis of typical bidding patterns induced by various negotiation strategies. It is the objective of this paper to propose a method and some metrics that facilitate a precise characterization of the negotiation dance. In turn, such a characterization of the dynamics of negotiation may contribute to the identification of explanations for such findings. It is the aim of this paper to at least partially identify some of the reasons that may explain particular findings, that is, to associate particular aspects of a negotiation problem or strategy with particular extreme values (e.g., minimum or maximum) of the metrics defined below.

We illustrate the use of these concepts for the analysis of concession tactics. For example, although it is generally acknowledged that a concession should actually increase the utility of the opponent and not just be a move that decreases one's own utility, in practice, as we will show, such behaviour is not always achieved by strategies that have been designed to concede towards the opponent. Moves that reduce both the agent's own as well as its opponent's utility have been called unfortunate moves (cf. [1]). Both humans as well as software agents using the ABMP strategy were observed to make such moves in negotiation experiments reported in [1], but humans made fewer of them.

Another aspect that plays a central role in DANS is the analysis of the domain of negotiation. In all informal literature on negotiation it is stressed that the negotiator should prepare with respect to the domain, the opponent, and the negotiator's own preferences. However, the literature on automated negotiations, the aspect of the domain leaves room for improvement. Some strategies use domain knowledge, but no formal analysis has been made of domains in terms of characterizing properties. In this paper we introduce a number of characterizing properties that proved useful in understanding the strengths and weaknesses of bidding strategies.

The analytical method that we propose in this paper combines the standard analysis described in [16] and the analysis of negotiation dynamics of [8] to give a better understand-

ing of a negotiation strategy performance. The method takes into account negotiation factors from [10] that influence the negotiation performance to gain a better understanding of when a negotiation strategy is applicable.

The paper is organized as follows. The next section discusses related work. Section 2.4 discusses negotiation factors that are included in the proposed analysis method. In Section 2.5, we briefly introduce the topic of negotiation dynamics. Section 2.6 introduces the move-based analysis method and some metrics for analyzing dynamic negotiation properties. Section 2.7 explains phases of the analysis method. In Section 2.8, the method is illustrated by analyzing the Trade-Off [4], ABMP [11], and Bayesian learning [9] strategy in various negotiation domains. Finally, the paper concludes with some suggestions for research on automated negotiation derived from the proposed analysis method.

2.2 Related Work

The scope of the current paper concerns the negotiation dynamics as a pattern of offers (cf. [15]). That is, our work concerns bargaining, a method for reaching joint agreements by means of exchanging offers according to e.g., an alternating offers protocol. With the exception of [1] and [2] all papers in the literature that discuss the quality of negotiation strategies, concern outcome analysis, not the negotiation dance.

This section starts with a summary of measures used for outcome analysis. This work is used within DANS and is complementary to the work presented in later sections of this paper to analyse of dynamics of negotiations.

In order to develop efficient negotiation strategies that are robust against and outperform other strategies, it is important to be able to evaluate the dynamic behaviour induced by negotiation strategies. Therefore, Section 2.3 focuses on the literature regarding aspects useful for the analysis of dynamic patterns of negotiations.

2.2.1 Optimal Solutions and Performance Metrics

The most common outcome performance metrics (see Figure 2.1) used to determine the quality of an agreement with respect to each of the players include the distance to the Pareto Efficient Frontier, the Nash Product, and the Kalai-Smorodinsky outcome, see e.g., [2] and e.g., [15] for formal definitions. Other global measures taken are the (average) number of negotiation rounds R needed to reach an agreement, the number of agreements A reached in a tournament, and the time T taken by each party.

Pareto frontier. A bid is Pareto efficient if given a set of alternatives, no movement from the bid to an alternative exists that can make at least one individual better off without making any other individual worse off. Typically, there exist multiple Pareto optimal solutions. The set of such solutions is called a Pareto frontier.

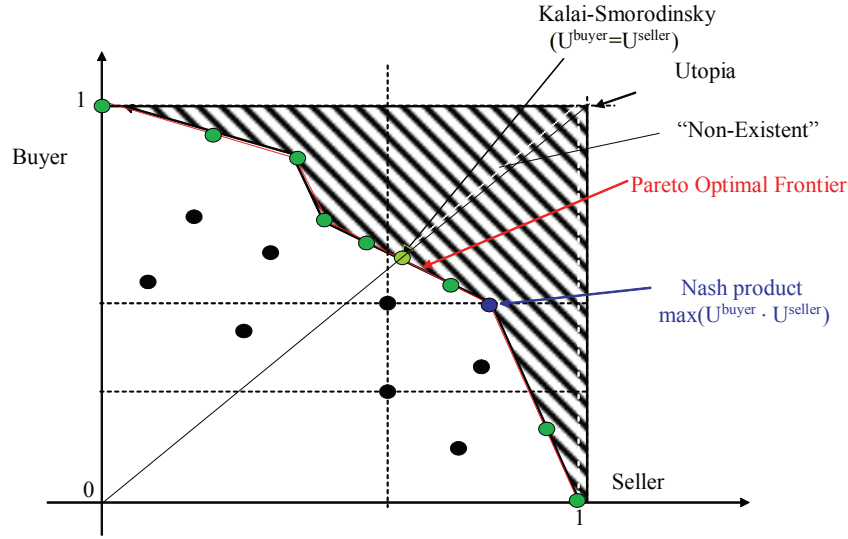


Figure 2.1: Standard Analysis

Nash product. The Nash product is that outcome that maximizes the product of the utilities of the parties. Nash product satisfies certain axioms. It is an invariant to affine transformations, independent to irrelevant alternative. Nash solution is always Pareto optimal. In addition, Nash solution is symmetrical, meaning that if both players have the same utility functions, then symmetry demands that both get equal payoffs.

Kalai-Smorodinsky. The Kalai-Smorodinsky outcome is that point on the Pareto frontier which maintains the ratios of maximal gains. In other words, assuming that the utility functions of the parties are normalized and map into the interval $[0; 1]$, the Kalai-Smorodinsky solution is a Pareto efficient outcome that has equal utilities of the two parties.

The authors of [12] add the following properties to the usual outcome properties:

- Social welfare: the sum of the utilities of the negotiators for the agreement should be as high as possible.
- Invariance: the solution is invariant under the application of positive affine transformations on the utility functions of both agents.
- Independence of irrelevant alternatives solutions.

Nongaillard and co-authors consider not only the here mentioned utilitarian social welfare, but also egalitarian social welfare, see [22].

In [13], a classification scheme is provided that defines some properties that are oriented towards rationality and the use of resources. They identify the following desirable properties for a negotiation protocol and strategies: computational efficiency, communica-

tion efficiency, individual rationality, distribution of computation, Pareto efficiency, and symmetry of power between agents. They provide characteristics useful for negotiation system design:

- Cardinality: number of issues, and one-to-one, many-to-one, many-to-many negotiators.
- Agent characteristics: the role it plays (buyer, seller, intermediary), its rationality (perfect or bounded), its knowledge about other agents' preferences, its social behaviour (self-interested vs. altruistic), and its bidding strategy.
- Information parameters: the value of goods (public / private), the nature of goods (discrete / continuous), price quotes, and transaction history.
- Event parameters: validity of bids, visibility of bids (not for one-to-one negotiations), clearing schedule with allocation parameters, timeouts, and a quotes schedule.

2.3 Negotiation Dance Literature

In [1] and [2], a formalization of the negotiation process is provided together with a set of performance properties that facilitate evaluation of the quality of the agreement reached, based on the work of [16] and [17]. The paper also discusses some dynamic properties of the bidding. The authors used the SAMIN system to analyze the ABMP strategy of [11] playing against itself and playing against human negotiators. The experiments showed that human and ABMP negotiators primarily made concession moves (see for a precise definition Section 2.6.1). Additionally, it was shown that humans were more diverse, i.e. the types of negotiation moves they performed were more diversified.

In [8] it has been shown that the performance of a negotiation strategy might depend on the negotiation domain and preference profiles of the negotiation parties. The analysis methodology, therefore, should include a mechanism to vary all these factors influencing the negotiation behaviour. The analysis method proposed already includes the following factors: size of the negotiation domain, predictability of the preferences, opposition of preferences, and negotiation strategy of the opponents. The paper shows that these factors can influence learning of opponent's preferences and, as a result, negotiation performance of a strategy.

The initial, informal classification of negotiation moves and the results reported in [1] form the inspiration of the current paper. In combination with the belief expressed by many that the pattern of offers exchanged influences the negotiated outcome, see e.g. [15], this motivated our study of negotiation strategies from the perspective of the negotiation dynamics and the actual moves made.

On the basis of the literature and our own experience we conclude that an analysis of negotiation dynamics requires the use of both theoretical as well as experimental evaluation

methods, in which at least the following aspects are attended to:

- competition with other strategies and itself,
- case studies of varying complexity,
- domains with various characteristics, and
- theoretical properties of the dynamics.

For papers that focus on the competition with other strategies the reader is referred to e.g., [4, 6, 17]. The next sections discuss the value of the other aspects.

2.4 Negotiation factors

Negotiation always takes place in a setup defined by the negotiation domain and the preference profiles of the parties involved. In [8] it has been shown that performance of the negotiation strategies depends on the negotiation setup. Therefore, we identify a number of factors that can influence a negotiation strategy or one of its components. A number of negotiation factors influencing negotiation behaviour have been reported in [8] and [10]. We reuse these factors in our method.

Size of the negotiation domain. Complexity of the negotiation domain and preference profiles is determined by the size of the negotiation domain. Size of the domain can influence learning performance of the negotiation strategy and, thus, the outcome reached by the strategy [10]. The size of the domain is exponential with respect to the number of issues. Therefore, the experimental setup in the analysis method should have a set of domains ranging from low number of issues to higher number of issues.

Predictability of the preferences. Negotiation strategies can try to exploit the internal structure of the preferences in order to improve one's own efficiency. I.e., the Trade-off strategy proposed in [4] assumes that distance measures can be defined using domain knowledge for the preferences of the opponent. These measures combined with the opponent's offers allow the Trade-off strategy to predict opponent preferences and as a result improve efficiency of the bidding. In [8], however, it has been shown that in case of a mismatch of the domain knowledge and the actual structure of the opponent's preferences the performance of a strategy can drastically drop. Therefore, we introduce the notion of the predictability of the preferences into our method.

Issues are called predictable when even though the actual evaluation function for the issue is unknown, it is reasonable to expect some of its global properties. For example, a price issue typically is predictable, where more is better for the seller, and less is better for the buyer, and the normal ordering of the real numbers is maintained; an issue concerning colour, however, is typically less predictable.

Opposition of the preferences. The results of analyzing negotiation dynamics presented in [8] revealed that some negotiation strategies are sensitive to preference profiles with

compatible issues. Issues are compatible if the issue preferences of both negotiating parties are such that they both prefer the same alternatives for the given issue. Negotiation strategies may more or less depend on whether preferences of the negotiating parties are opposed or not on every issue. That is, using some strategies it is harder or even impossible to exploit such common ground and agree on the most preferred option by both parties for compatible issues (humans are reported to have difficulty with this as well; cf. [21]). A selection of preference profiles should therefore take into account that both preference profiles with and without compatible issues are included.

2.4.1 Negotiation domains and preference profiles

Ideally, negotiation domains used in the analysis should cover all range of the factors sketched above. The selection of the domains presented in this paper is not intended to cover all variations of the domain factors influencing the negotiation performance. The negotiation domains used in this paper are:

The *Second hand car selling* domain, taken from [11], includes 5 issues. Only the buyer's preferences and the price issue are predictable, in the sense that an agent can reliably predict the other agent's preferences associated with an issue.

The *Service-Oriented Negotiation* domain, taken from [4], includes 4 issues. All issues are predictable, i.e. based on available "domain knowledge" preferences can be reliably predicated. The preference profiles have the strongest opposition in our setup.

The *AMPO vs City* domain, taken from [15], includes 10 issues, of which only 8 are predictable. This is the biggest domain in our experimental setup. Information about the opponent's issue priorities is not available, i.e., the weights agents associate with issues.

2.4.2 Negotiation strategies

The following strategies have been studied: The ABMP strategy [11], a concession oriented strategy, which computes bids to offer next without taking domain or opponent knowledge into account. (Experiments were run with a negotiation speed of 0.1 and a concession factor of 1, see [11].)

The *Trade-off* strategy is based on similarity criteria [4], and exploits domain knowledge to stay close to the Pareto Frontier. The "smart" version of this strategy performs nice moves if possible; otherwise it concedes a fixed amount 0.05 (cf. [4]). For the Service-Oriented Negotiation domain, we reproduced the results presented in [4].

The *Random Walker* strategy randomly jumps through the negotiation space, and can be run with or without a break-off point (to avoid making offers below that utility). Random Walker serves as a "baseline" strategy. This strategy has also been called the *Zero Intelligence* strategy [5].

A negotiation strategy that uses a learning technique based on the Bayesian learning algorithm proposed in [9]. The opponent model in [9] is based on learning a probability distribution over a set of hypotheses about evaluation functions and weights of issues. The probability distribution is defined over the set of hypotheses that represent an agent's beliefs about an opponent's preferences. Structural assumptions about the evaluation functions and weights are made to decrease the number of parameters to be learned and simplify the learning task.

The authors propose two versions of the learning algorithm. In the first version of the algorithm each hypothesis represents a complete utility space as a combination of weights ranking and shapes of the issue evaluation functions. The size of the hypotheses space growth exponentially with respect to the number of issues and thus is intractable for negotiation domains with a high number of issues (more than 6 issues, where each issue has 10 or more possible values).

The second version of the algorithm is a scalable variant of the first one. This version of the agent aims to learn a probability distribution over the individual hypotheses about the weight and shape of the issue evaluation function independently of other issues. The computational tractability of the learning is achieved by approximating the conditional distributions of the hypotheses using the expected values of the dependant hypotheses.

2.5 Negotiation Dynamics

In the analysis of negotiation strategies, not only the outcome of a negotiation is relevant, but also the bidding process itself is important. Mistakes made during the bidding can have an enormous impact on both players. Although experienced negotiators confirm this, and it is also recognized by researchers of negotiation strategies for automated negotiation, this hypothesis is difficult to quantify. Examples from human negotiations are of the form: “a wrong offer can upset relationships, even causing the other party to walk away”, or “Sometimes an offer that is meant as a concession to the other party confuses the issues. This can only be circumvented if there is enough trust between the parties to exchange some information on their respective preferences.”

From the point of view of automated negotiation, the objective is to stay as close as possible to the Pareto Efficient Frontier. However, in automated negotiations typically no prior information is exchanged about the preferences of the negotiating parties, and none of the players know where the Pareto Efficient Frontier actually is. It thus remains a challenge to stay or end close to that Frontier. To this end, opponent modelling may be used to predict which bids will be appreciated by the other party, see, e.g., [3] and [14].

More precisely, five key factors can be identified that shape the outcome of a bilateral negotiation with incomplete information:

- knowledge about the negotiation domain,

- one's own and one's opponent's preferences,
- process attributes (e.g. deadlines),
- the negotiation strategies, and
- the negotiation protocol.

In this paper, our interest is in analyzing, classifying and in precisely characterizing aspects of the negotiation dynamics that influence the final agreement of a negotiation. The main interest thus is in proposing concepts and metrics that relate these factors to specific aspects of the negotiation dynamics and to thus gain a better understanding of the final outcome of a negotiation. The analysis does not take the features of the protocol per se into account but instead focuses on the exchange of offers. In principle, the method allows for generalizations to multi-party negotiations but we do not consider such extensions here. More information on multi-party negotiations can be found in e.g. [17].

2.6 Step-Wise Analysis

In bilateral bargaining, the negotiation dynamics is completely represented by the sequence of *offers* $t = \langle b_S^1, b_O^2, b_S^3 \dots \rangle$ exchanged between parties S and O , also called the *negotiation trace*. A negotiation trace is called *closed* if it ends in either an *accept* or *withdraw* move by either party. In this section the basic notions of the step-wise analysis method are defined: classes of negotiation moves, metrics, outcome properties, and patterns over negotiation moves. After that the different phases of the method are defined.

2.6.1 Negotiation Moves

The key unit of negotiation dynamics analysis is a single negotiation step performed by one of the negotiating parties. A negotiation step in bargaining consists of an offer proposed by one party to the other. If this offer is not the first offer proposed by an agent, it typically is computed using at least the previous offer proposed by that agent as input. To record this fact and to facilitate notation below, formally, a negotiation step s by agent a is modeled as a transition from a previous offer b_a to a newly proposed offer b'_a , which is written as $b_a \rightarrow b'_a$. Such moves can be classified based on the associated utility for both parties.

Bosse and Jonker [1] introduced 4 classes of steps: concession, unfortunate, fortunate and selfish steps. Firstly, negotiation strategy should be able to make *concessions*. In a concession, an agent trades in own interest in favour of the opponent to reach an agreement. Unfortunately, in a closed multi-issue negotiation such a move can lead to an alternative that is even worse for the opponent due to lack of information about the opponent's preferences. Such move is called *unfortunate*. In addition, the agent can go up with respect its own and down with respect to the opponents utility function. Such a move is called a

selfish move. The last class is called the class of *fortunate* moves, i.e., a move towards the Pareto Efficient Frontier, going up with respect to the utility functions of both parties.

For the step-wise analysis method the classes of negotiation moves of [1] have been extended with two additional classes. In a number of papers, it has been suggested, that it is smart not to make concessions too soon, but to move over one's iso-utility lines first [4]. For this reason, a separate category of *nice* moves that move in the direction of the opponent but do not concede with respect to the agent's own utility is introduced. An example of a strategy that is designed to make such moves is the Trade-Off Strategy based on similarity criteria discussed in [4], a variation is proposed in [12]. Additionally, so-called *silent* moves are introduced to represent the fact that parties sometimes repeat their offers, and do not make any concessions at all such as in a Boulware or take it or leave it strategy, see [11].

Each type of move in a negotiation typically has a distinct role or function, though in automated negotiation systems not all of these types of move are taken into account. Fortunate moves happen spontaneously in human negotiations (see [1]). Having a strategy that is able to perform such moves deliberately is beneficial, since such moves can be used to recover from moves away from the Pareto Efficient Frontier, e.g., as the result of concessions or unfortunate moves. The latter two moves aim at reaching a jointly acceptable outcome. Although in general, it would be best to avoid unfortunate moves when conceding, it is impossible to guarantee this when Opponent's preferences are not completely known. Selfish moves may be performed by an agent to signal to the other party that a previous move is not appreciated. The role of nice and silent moves has been discussed above.

Before formally defining the concepts below, some additional notation is introduced. $U_S(b)$ denotes the utility of "Self" with respect to bid b . Similarly, $U_O(b)$ denotes the utility of "Opponent" with respect to b . We use $\Delta_a(b, b') = U_a(b') - U_a(b)$, $a \in S, O$, to denote the utility difference of two bids b and b' in the utility space of agent a . We also write $\Delta_a(s)$ to denote $\Delta_a(b, b')$ for a move $s = b \rightarrow b'$. Here we present a precise definition of the classes of negotiation moves proposed in [1] extended as discussed above. These move categories define the core of the move-wise analysis method.

Definition 1 (Move Classes). *Let $s = b_S \rightarrow b'_S$ be a move in the bidding by Self (the definition for Opponent is completely symmetric). Then the negotiation move s taken by Self is classified as a:*

1. *Fortunate Move, denoted by $(S+, O+)$, iff: $\Delta_S(s) > 0$, and $\Delta_O(s) > 0$.*
2. *Selfish Move, denoted by $(S+, O)$, iff: $\Delta_S(s) > 0$, and $\Delta_O(s) = 0$.*
3. *Concession Move, denoted by $(S-, O)$, iff: $\Delta_S(s) < 0$, and $\Delta_O(s) = 0$.*
4. *Unfortunate Move, denoted by $(S, O-)$, iff: $\Delta_S(s) = 0$, and $\Delta_O(s) < 0$.*
5. *Nice Move, denoted by $(S=, O+)$, iff: $\Delta_S(s) = 0$, and $\Delta_O(s) > 0$.*
6. *Silent Move, denoted by $(S=, O=)$, iff: $\Delta_S(s) = 0$, and $\Delta_O(s) = 0$.*

The proposed classification is exhaustive, and all move classes are disjoint. (To allow for some marginal errors the areas of the Nice and Silent moves can be stretched somewhat in the analysis. In that way, a move in which only 0.005 of Self's utility is lost would still be classified as e.g. Nice, instead as a Concession.) In a concession move some own utility needs to be conceded but Opponent's utility may stay the same. In such cases, Self can claim that it made a concession move by arguing that it conceded some of its own resources.

2.6.2 Step Metrics and Pattern Properties

Having established different types of negotiation moves that are useful in the analysis of negotiation strategies, we now introduce and define metrics in terms of these moves that can be used for the analysis of negotiation traces. First, some additional notation is defined. Given a trace $t = \langle b_S^1, b_O^2, b_S^3 \dots \rangle$ of offers, t_i denotes the i 'th element of this sequence. Let t_S (resp. t_O) denote the sequence of moves from t that are made by agent "Self" (resp. "Other") and let class $c \in \text{Fortunate, Nice, Concession, Selfish, Unfortunate, Silent}$; then t_c denotes the subsequence of moves that belong to class c . Finally, $t_{\langle a, c \rangle}$, also written t_{ac} , denotes the subsequence of moves by $a \in S, O$ that belong to class c . The following move metrics are introduced here:

Definition 2 (Number of Moves per Trace). . The number of moves $\#t$ in a trace t of length $|t| = n$ is defined as follows: $\#t = |t| - 1$.

Definition 3 (Total Utility per Class). The pair $Total_c(t)$ of sums of utility differences in all moves of class c in a sequence t of moves is defined by:

$$Total_c(t) = (Total_{Sc}(t), Total_{Oc}(t)),$$

where for any agent $a \in S, O$: $Total_{ac}(t) = \sum_i \Delta_a(t_c^i)$.

Definition 4 (u -Average Utility per Class). The pair $u-Avec(t)$ of average differences in utility in all moves in class c in a sequence t of moves is defined by:

$$u-Avec(t) = (u-Ave_{Sc}(t), u-Ave_{Oc}(t)),$$

where for any agent $a \in S, O$:

$$u-Ave_{ac}(t) = \sum_i \Delta_a(t_c^i) / \#t_c.$$

Here $\#t_c$ is a number of moves of class c in trace t . This metric measures the average utility conceded per negotiation move. A relative measure can be defined in terms of this metric to identify how much utility has been conceded by agent a relative to the other, indicated by agent g : $u-Ave_c(t_a) / u-Ave_c(t_g)$. This figure, if not identical to 1, indicates that one party is a conceder relative to the other, and that concessions may not have been paced and linked to that of the other party, as is advised by Raiffa [16] (p. 128).

Definition 5 (% per Class). The percentage $\%_c(t)$ of class c moves in a trace t is defined by: $\%_c(t) = \#t_c / \#t$

Negotiation strategies can be designed with specific aims in mind that should be observ-

able as patterns in the negotiation dance. For example, the success of a strategy that is supposed to learn its opponent's preferences can be verified by checking whether the frequency and/or size of unfortunate moves over a negotiation trace decreases. Such patterns can be seen as a measure of adaptability of a party to the opponent. Another useful measure of the sensitivity to the opponent's preferences can be defined by comparing the percentage of fortunate, nice and concession moves that increase the opponent's utility to the percentage of selfish, unfortunate and silent moves that decrease it. Intuitively, the more an agent performs moves that increase its opponent's utility the more sensitive to the needs of its opponent, it is said to be.

Definition 6 (Sensitivity to Opponent's Behaviour). *The measure for sensitivity of agent a to Opponent's behaviour is defined for a given trace t by:*

$$BehavSens_a(t) = \frac{\%Fortunate(t_a) + \%Nice(t_a) + \%Concession(t_a)}{\%Selfish(t_a) + \%Unfortunate(t_a) + \%Silent(t_a)} \quad (2.1)$$

In case no selfish, unfortunate or silent moves are made we stipulate that $BehavSens(a, t) = \infty$. If $BehavSens_a(t) < 1$, then an agent is more or less insensitive to Opponent's behaviour; if $BehavSens_a(t) > 1$, then an agent is more or less sensitive to Opponent's behaviour, with complete sensitivity for $BehavSens(a, t) = \infty$. Typically, this sensitivity measure varies with different domains and different opponents and averages over more than one trace need to be computed. Note that the notion of sensitivity is *asymmetric*: one agent may be sensitive to its opponent's behaviour, but not vice-versa. In section 2.8, this metric is used to analyze the sensitivity of two existing negotiation strategies. Furthermore, its relation to the knowledge available to the agent of the opponent's behaviour and the negotiation outcome is discussed.

We expect that the Random Walker strategy to make about 25% of moves in each of the fortunate, unfortunate, selfish, and concession classes due to its randomized selection of bids. This strategy is included in our method as a benchmark strategy and is used to validate our measures (see Table 2.1).

2.6.3 Sensitivity to Opponent's Preferences

A successful negotiation strategy should search for offers that maximize the agent's own utility while increasing the chance of the acceptance by the opponent. We assume that a rational negotiating agent would more easily accept offers higher utility than those with lower utility. To increase the chance of an acceptance it is, therefore, rational to increase the opponent's utility of an offer without giving in with respect the agent's own interests by means of trade-offs between the issues. Ideally, such a search procedure would lead to offers on the Pareto Efficient Frontier. Thus, an efficient negotiation strategy should generate offers from the Pareto frontier or at least as close as possible to the frontier depending on the limitation of the strategy.

To generate offers that are close to the Pareto Efficient Frontier a negotiation strategy must use information about the opponent's preferences in addition to its own preferences.

Therefore, to assess the performance of a negotiation strategy we need to measure its sensitivity to the opponent's preferences. Given the assumptions of the opponent's rationality we define a measure of sensitivity as follows.

Definition 7 (Sensitivity to Opponent's Preferences). *To measure sensitivity to the opponent's preference we calculate the average difference between Opponent's utility of the bids generated by the strategy and utility of a bid on the Pareto frontier:*

$$PrefSens(t_s) = \frac{1}{\#t_s} \sum_{i=1}^{\#t_s} U_O(t'_i) - U_O(t_i) \quad t_i \in t_s$$

where t'_i is defined as

$$t'_i = \arg \max_{U_S(b)=U_S(t_i)} U_O(b), b \in D$$

Figure 2.2 visualizes the sensitivity to Opponent's preferences measure. The figure puts all possible negotiation outcomes in a given negotiation domain on a two-dimensional space. Each dimension represents utility of an offer respectively to the negotiating parties. A negotiation strategy that is perfectly sensitive to Opponent's preferences would propose Pareto efficient offers only and, therefore, would have $PrefSens(t) = 0$. The higher the measure the worse sensitivity of the strategy.

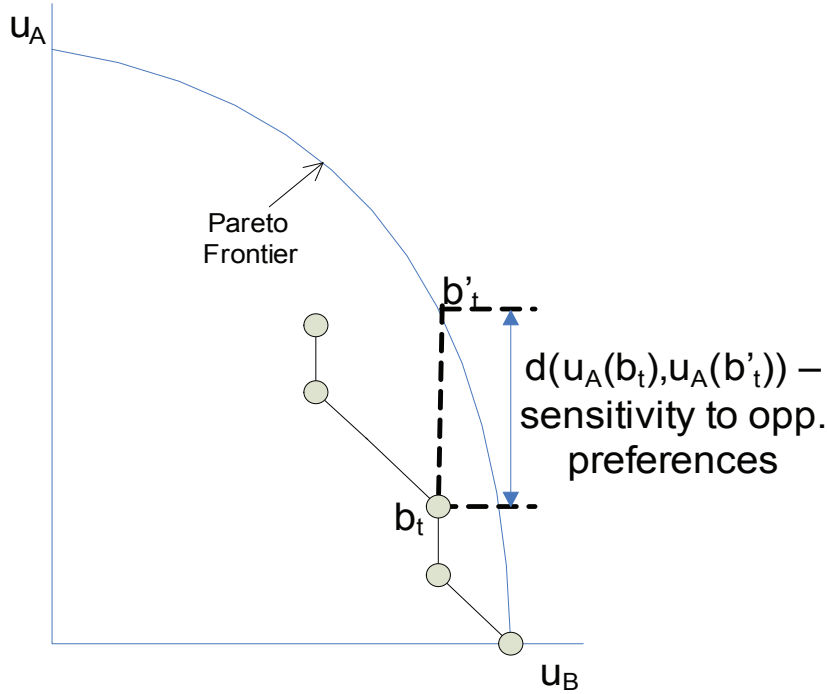


Figure 2.2: A Negotiation Problem and Optimal Solutions

As said before, a strategy that is sensitive to the opponent's preferences requires information on the opponent's preferences. The ABMP strategy belongs to the class of the

Time-Dependent Tactics that do not consider the opponent's preferences. Given the definition of the measure of sensitivity to the opponent's preferences, we can expect low sensitivity of the ABMP strategy (see Table 2.5).

A typical solution to increase the sensitivity to the opponent's preferences is to use an opponent model. The Bayesian strategy uses a learning algorithm to guess the opponent's preferences and tries to bid on the Pareto Efficient Frontier to increase the chance of acceptance. Thus, the Bayesian strategy should score high on the sensitivity measure in case when learning is successful (see Table 2.5). As we saw in the previous section the performance of the Trade-Off strategy strongly depends on the negotiation domain and the opponent's strategy. A similar result is to be expected in the sensitivity of that strategy to the opponent's preferences. The Random Walker does not consider the preferences of the opponent at all and would therefore have low sensitivity.

2.7 Negotiation Analysis Methodology

A methodology for analyzing the performance of various strategies for automated negotiating agents should include the by now standard metrics based on solution concepts such as the Nash product as well as the metrics introduced above to analyze the negotiation moves to reach an agreement. The need to include metrics related to the negotiation dance itself is clear from the performance of a Random Walker negotiating agent. As we show in Section 2.8, even a Random Walker can obtain an outcome close to the Pareto Efficient Frontier and good for the Random Walker provided that its opponent uses a reasonable negotiation strategy. A proper analysis should reveal that such results need to be contributed to the strategy or performance of the opponent rather than to the Random Walker agent. Though this may seem obvious from the performance of a Random Walker in any given negotiation for other negotiation strategies similar conclusions may only be reached by analyzing the performance of that strategy in various domains and in combination with different preference profiles. A methodology for analyzing the performance of a negotiating agent thus should involve a careful setup of experiments as well as a range of different metrics to be able to reach sound conclusions about the agent's performance.

The methodology that we propose consists of seven steps to facilitate the setup of a tournament for analyzing negotiation strategies. The first four steps of the proposed method define the tournament whereas the last three steps concern the analysis of the results. Although the method proposed does not provide any guarantees that the proper results will be obtained, it does facilitate and structure the process of setting up negotiation experiments to obtain such results. Moreover, by reiterating the process the results of previously performed experiments can be used to refine the experimental setup and may suggest variations not initially considered.

2.7.1 Phases of the Negotiation Analysis Method

Based on the above concepts, the analysis method is specified by:

- 1: Starting point: the strategies to be analysed, and a library of domains and of other strategies that can be used to test the input strategies.

In Section 2.3 we introduced a number of criteria for selection of negotiation domains, preference profiles and opponent strategies. Ideally, then, one would use an experimental setup based on random sampling of the domains and profiles in order to deal with this problem. However, it is not clear how to setup such a sampling procedure. Therefore, we selected a number of negotiation domains to be used in our experimental setup.

The opponent's negotiation strategy is one of the negotiation factors influencing the agent's negotiation performance [8]. It is important, therefore, to be able to include a wide range of existing negotiation strategies.

- 2: If necessary, implement the input strategies.

The proposed method is based on the analysis of empirical data and, therefore, requires an implementation of the input negotiation strategies that can be run in an experimental setup. Ideally, the input strategy should be implemented in the same environment with the strategies and negotiation domains and profiles selected in the 1st step. Otherwise, a communication between the strategies must be established and the negotiation domains and profiles must be translated and made available for the input strategies.

- 3: Set up a tournament with the selected negotiation strategies and case studies.

A tournament a la [6] is used to experiment with various strategies. In the tournament, strategies play against each other, against themselves, and are applied to varying negotiation domains, with varying preference profiles. Multiple negotiation sessions of a single negotiation setup should be run in the case of non-deterministic negotiation strategies in the tournament.

- 4: Run the tournament and log every negotiation.

The negotiation log should include information about the tournament and settings of an individual session: the names of the strategies, the domain name, and the preferences of the players. Furthermore, it is important to produce an extensive log with all information needed to calculate the proposed metrics. Therefore, the log should include bids proposed by every negotiating party in a negotiation session and their utilities.

- 5: Calculate defined metrics.

Given the logs produced in the previous step, the following metrics introduced in this paper should be calculated for every negotiation session:

- utilities of the negotiation outcomes;
- number of negotiation moves;
- number of moves, %, and total utility per class;
- sensitivity to opponent's behaviour;
- sensitivity to opponent's preferences;

6: Apply statistics and produce analytical results.

In the case of non-deterministic negotiation strategies in the tournament as explained in the step 3, average values of the metrics should be calculated and used for analysis.

7: Interpret results, produce graphics.

The goal of the proposed method is to assess the negotiation performance of a negotiation strategy given a number of metrics of the negotiation outcome and dynamics. The assessment is based on the interpretation of the results received from the variation of the values of the negotiation factors introduced in this paper.

Graphics such as the two-dimensional utility plots of the negotiation outcome space (see, for example, Figure 2.2) can help to find interpretations for the results. They highlight the dynamic properties of the negotiation traces proposed here as well as the standard analysis measures.

The proposed analyses method was applied in an experimental setup using the open negotiation environment for heterogeneous negotiating agents, presented in [7]. The environment makes use of advanced software engineering technologies allowing simple and quick integration of existing negotiation strategies. The framework provides a simple application programming interface and a number of auxiliary common services to the agent to simplify the task of strategy implementation. It has a rich library of implemented negotiation strategies.

The analytical module of the environment can be easily extended with new metrics and has advanced logging functionality. Logs of the environment can be exported to widely-used analytical software, such as Excel and Mathematica. The environment has tools to generate a setup for a tournament and run it producing logs with the corresponding information.

2.8 DANS Applied

This section illustrates the DANS analysis method for a combination of strategies and negotiation domains, while focusing on individual moves $\tilde{U}$ in particular *unfortunate moves* $\tilde{U}$ and on the sensitivity of four negotiation strategies with respect to preferences and

Profile	High sensitivity	Low sensitivity
BehavSens	∞	0
PrefSens	0	1

Table 2.1: Extreme values for the sensitivity measures.

behaviour of the opponent. Table 2.1 reminds how the sensitivity values should be interpreted and give extreme values for the high and low sensitivity.

A tournament with the strategies and domains of the previous section was set up and run. A full analysis was made of the type of moves made, which was then used to calculate the average sensitivity ratio's for all tested strategies over multiple runs against all strategies (including itself) in the domains described above.

2.8.1 Outcome Analysis

The traditional (and valuable) method for analysis of strategies in terms of the outcome and its properties is applied first to the chosen combination of strategies and negotiation domains as explained in the previous section.

Negotiation Domain	Strategy of Agent A	Strategy of Agent B															
		ABMP				Trade-Off				Random Walk (break-off=0.6)				Bayesian			
		Utility	Distance to			Utility	Distance to			Utility	Distance to			Utility	Distance to		
		Agent A	Agent B	Nash	Kalai	Agent A	Agent B	Nash	Kalai	Agent A	Agent B	Nash	Kalai	Agent A	Agent B	Nash	Kalai
Car domain	ABMP	0.83	0.93	0.08	0.09	0.90	0.83	0.10	0.09	0.86	0.76	0.18	0.17	0.82	0.96	0.10	0.11
	Trade-Off	0.72	0.94	0.19	0.20	0.84	0.92	0.08	0.08	0.87	0.76	0.19	0.17	0.84	0.94	0.08	0.08
	Random	0.78	0.83	0.17	0.17	0.74	0.89	0.18	0.18	0.73	0.76	0.26	0.25	0.74	0.86	0.19	0.19
	Bayesian	0.78	0.96	0.14	0.15	0.87	0.94	0.04	0.05	0.74	0.85	0.19	0.19	0.89	0.92	0.03	0.03
	Average	0.91	0.15	0.15		0.90	0.10	0.10		0.78	0.21	0.20		0.92	0.10	0.10	
AMPO vs City	ABMP	0.73	0.63	0.16	0.11	0.70	0.59	0.19	0.15	0.58	0.65	0.18	0.18	0.65	0.70	0.10	0.09
	Trade-Off	0.62	0.53	0.27	0.23	0.74	0.73	0.06	0.01	0.67	0.58	0.21	0.17	0.65	0.70	0.10	0.09
	Random	0.69	0.68	0.10	0.07	0.54	0.60	0.25	0.24	0.65	0.65	0.15	0.12	0.53	0.72	0.18	0.20
	Bayesian	0.65	0.70	0.10	0.09	0.74	0.72	0.07	0.01	0.68	0.50	0.28	0.24	0.73	0.70	0.08	0.03
	Average	0.63	0.16	0.12		0.66	0.14	0.10		0.60	0.21	0.18		0.70	0.12	0.10	
Service Oriented Negotiation	ABMP	0.51	0.59	0.36	0.33	0.65	0.49	0.38	0.32	0.73	0.66	0.19	0.13	0.76	0.84	0.02	0.06
	Trade-Off	0.52	0.65	0.31	0.29	0.79	0.77	0.09	0.02	0.55	0.61	0.31	0.28	0.76	0.82	0.03	0.05
	Random	0.63	0.60	0.28	0.23	0.63	0.77	0.14	0.15	0.66	0.70	0.17	0.14	0.78	0.78	0.08	0.00
	Bayesian	0.79	0.75	0.11	0.03	0.81	0.71	0.15	0.08	0.80	0.74	0.12	0.05	0.76	0.82	0.03	0.05
	Average	0.65	0.26	0.22		0.68	0.19	0.14		0.68	0.20	0.15		0.82	0.04	0.04	

Figure 2.3: Result of Standard Analysis

Table 2.3 presents the results of the outcome analysis. The average values are calculated over all opponents' strategies per negotiation domain and role in the domain. The ABMP strategy shows excellent performance (average utility of 0.85 for Role A and 0.91 for Role B) on the Car domain for which it was designed. The Trade-Off strategy underperforms the ABMP strategy on this domain with average utilities of 0.82 and 0.90 for the roles A and B respectively. The Bayesian agent performs somewhat better than the Trade-Off strategy (average utility of B 0.82 for Role A and 0.92 for Role A). Random Walker is definitely a layman in this domain.

In the SON domain the Bayesian agent clearly outperforms the other strategies (average utility of 0.79 for Role A and 0.82 for Role B). Interestingly, the Trade-Off strategy performance varies from one opponent to another. It does well when playing against itself

(utility of 0.79 for Role A and 0.77 for Role B) or against Bayesian agent (utility of 0.76 for Role A and 0.71 for Role B) but fails against ABMP (utility of 0.52 for Role A and 0.49 for Role B) and the Random Walker (utility of 0.55 for Role A and 0.67 for Role B). Unfortunately, the ABMP strategy is not able to take advantage of the Trade-Off strategy but leaves better alternatives on the table. The Random Walker shows a performance that is on average better than that of the Trade-Off and ABMP strategies (average utility of 0.67 for Role A and 0.68 for Role B). However, this might be due to the selected reservation value of Random Walker (every bid will have a utility of at least 0.6). Setting the reservation value even higher would lead to even better results.

The results in the AMPOvsCity domain are similar to that of the SON domain. The distance of the negotiation outcome reached by the Bayesian strategy and the Nash and Kalai-Smorodinsky outcomes is bigger than in the SON domain. This can be explained by the significantly larger size of the AMPOvsCity domain that causes some degradation in the learning performance of the Bayesian strategy, see [10]. Due to the large size of the AMPOvsCity domain the Random Walker and the ABMP strategy show the worst performance among the other strategies.

The standard analysis method that focuses on the final outcome does not explain why such a performance is observed, e.g., the Trade-Off and the ABMP strategy underperform against the Random Walker strategy. Therefore, we propose the negotiation dynamics analysis method to get more information from the experimental results and get insights into the negotiation process.

2.8.2 Sensitivity to Opponent's Behaviour

Theoretically, over all domains and against all strategies, Random Walker would have a sensitivity value of 1. As expected, the Random-Walker shows approximately equal percentage of moves for the unfortunate, fortunate, selfish, and concession classes. It makes almost no nice or silent moves. Small deviation of the results can be explained by structural features of a negotiation domain.

ABMP shows an overall BehavSens of 4.15, Trade-off 4.13, Bayesian 6.21 and Random Walker 1.08. Note, that the efficiency of the agreement (see previous section) does not always correlate strongly with the BehavSens values of the strategies. The sensitivity scores of ABMP and Trade-Off can be better understood by considering the domains in which they played. Figure 2.4 shows typical runs in the AMPOvsCity domain.

Figure 2.4a shows a run of Trade-Off, representing the City, versus Random Walker (with a break-off set to 0.6), playing AMPO. The Random Walker strategy is insensitive with respect to the opponent's behaviour (BehavSens is always approximating 1) and strategy selects an offer in a random way. The Trade-off strategy uses the opponent's offers to select counteroffers and expects some rational behaviour of the opponent's strategy. This fact causes the unfortunate moves produced by the Trade-Off strategy (36% of all moves for Role A and 37% for Role B). This results in a low BehavSens of the Trade-Off strategy

in negotiation against Random Walker (1.04 for Role A and 1.33 for Role B), see the columns for Trade-Off in Table 2.5. However, the Trade-Off strategy in negotiations against itself and the Bayesian strategy makes significantly less unfortunate moves due to more rational and efficient behavior of the opponent and, as a result achieves much higher BehavSens scores (against itself: 5.25 for Role A and 6.14 for Role B; against Bayesian: 8.09 for Role A and 6.83 for Role B). In general, Trade-Off has high BehavSens in this domain (average value 4.07 for Role A and 3.89 for Role B)

Figure 2.4b shows Trade-Off (as City) vs ABMP (as AMPO) in which ABMP makes mostly concessions and silent moves by following its concessions tactic regardless of the moves of the opponent. It has rather low BehavSens (average value 1.56 for Role A and 2.93 for Role B) in this domain. ABMP shows more rational behaviour than Random Walker and, therefore, in this domain Trade-Off really exploits the available domain knowledge. The percentage of unfortunate steps in this negotiation drops to 27% for Role A and 32% for Role B.

Figure 2.4c shows Random Walker (City) vs ABMP (AMPO). ABMP in principle concedes on all issues, determining the size of the concession on the difference between the utilities of its own bid and that of its opponent. Unlike the Trade-Off strategy it does not use previous bids of the opponent to get insight into the opponent's preferences and as a result does not adapt much to the strategy of the opponent. Such a strategy will make unfortunate steps in case there are issues with compatible preferences (concession on such an issue would decrease utility of the opponent). AMPOvsCity does not have any issues with compatible preferences and, thus, ABMP does not make unfortunate steps. Given opposed preferences for every issue such a strategy typically produces concession moves (see Figure 2.4b,c). This results in a relatively high BehavSens in negotiations against Random Walker compared to that of the Trade-Off strategy (average value 1.47 for Role A and 1.89 for Role B). BehavSens for the ABMP strategy increases in negotiations against Trade-Off and Bayesian but has relatively smaller values than that of the Trade-Off strategy (against Trade-Off: 2.13 for Role A and 3.78 for Role B; against Bayesian: 1.70 for Role A and 4.56 for Role B).

Figure 2.4d shows Trade-Off (City) vs Bayesian (AMPO). The negotiation behaviour of the Bayesian strategy is very similar to that of the Trade-Off strategy: stays very close to the Pareto frontier and makes few unfortunate moves. The BehavSens of the Bayesian strategy is very high in this negotiation (13.29 for Role A and 6.69 for Role B). Unlike the Trade-Off strategy the Bayesian strategy does not rely on domain knowledge to generate offers that are close to the Pareto frontier. Instead it tries to learn the opponent's preference profile from the opponent's offers. The learned preference profile allows the Bayesian strategy to response to the opponent with more nice and concession moves and keep the number of unfortunate moves at minimum. As a result, it remains rather sensitive to the opponent's behaviour even when it negotiates against Random Walker (value 2.35 for Role A and 3.55 for Role B).

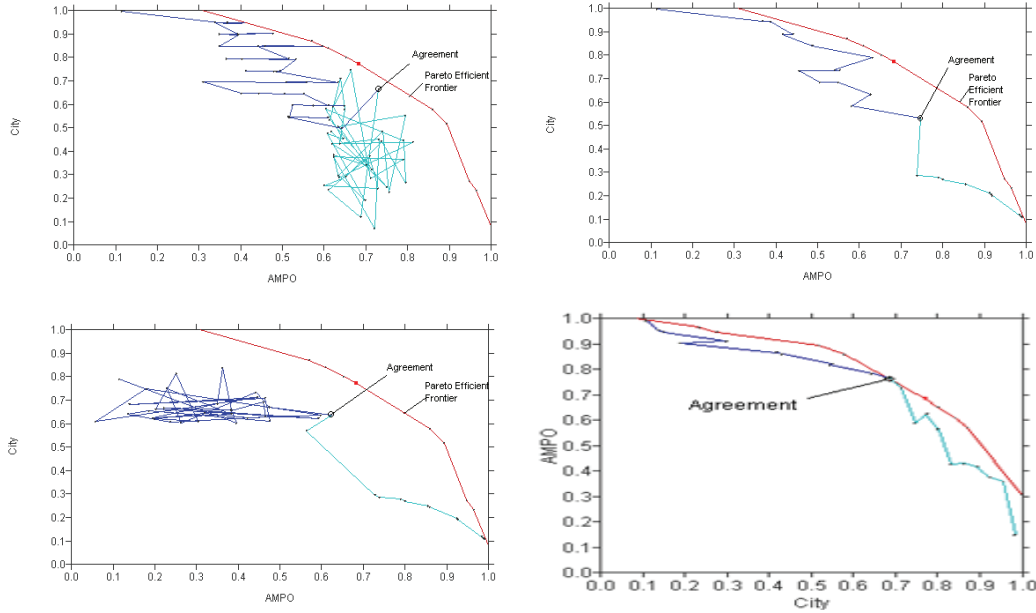


Figure 2.4: Dynamics of negotiation process for: a) Trade-Off (City) vs Random Walker strategy (AMPO), b) Trade-Off (City) vs ABMP strategy (AMPO), c) Random Walker (City) vs ABMP strategy (AMPO), d) Trade-Off (City) vs Bayesian strategy (AMPO). The Pareto Efficient Frontier is built according to its definition (see [16], pp. 227) using exhaustive search.

2.8.3 Sensitivity to Opponent's Preferences

The sensitivity to the opponent's preferences is measured in the same experimental setup as for the outcome analysis method and the step-wise method. Table 2.5 also presents the sensitivity results with respect to the opponent's preferences. As expected, the Bayesian strategy is the most sensitive to the opponent's preferences with average values per domain of $PrefSens \leq 0.09$. The Bayesian strategy is less sensitive when negotiating against the ABMP and Random Walker strategies than against the other strategies. This can be explained from the fact that the assumptions used in the learning algorithm do not hold for those strategies and the learned opponent's preference profile in these negotiations has lower quality.

The Trade-Off strategy is somewhat less sensitive to the preferences of the opponent (average $PrefSens \leq 0.22$ in the various domains) than the Bayesian strategy. The Trade-Off strategy can be efficient but has difficulties with domains with low predictability of the preferences. For example, it's sensitivity to the opponent's preferences is not as good in the 2nd Hand Car Selling domain ($PrefSens \leq 0.17$) as in the SON domain ($PrefSens \leq 0.13$) despite the fact that the later domain is much bigger than the earlier. The similarity functions for the 2nd Hand car domain often do not match the preferences of the opponent. In addition, the weights of the similarity function do not match the opponent's importance factors of the negotiation issues. The SON domain does not have

information about the weights of the similarity functions but the issues preferences perfectly match the similarity functions and thus the sensitivity to the opponent's preferences of the Trade-Off strategy increases.

The sensitivity to the opponent's preferences for the Trade-Off strategy depends on the strategy used by the opponent. In negotiations against ABMP and Random Walker the Trade-Off strategy is less sensitive to the opponent's preferences (e.g., in SON domain it has $PrefSens \leq 0.09$ in negotiations against Trade-Off and Bayesian whilst it has $PrefSens \leq 0.22$ in negotiations against ABMP and Random Walker). This variation can be explained by the fact that the Trade-Off strategy tries to match the opponent's preferences by maximizing the similarity of its offers with those of the opponent. If the opponent's offers are far from the Pareto frontier (such as in case of ABMP and Random Walker) the Trade-Off strategy would not be able to match the opponent's preferences.

The ABMP strategy shows rather robust performance but it is often outperformed by the smarter strategies. It has overall a rather low sensitivity to the opponent's preferences (average values over the various domains are $PrefSens \leq 0.42$). Still it is more sensitive than the Random Walker that, as expected, has the lowest sensitivity to the opponent's strategy (average values over the various domains are $Pref - Sens \leq 0.75$). Note, that the PrefSens value of the ABMP strategy does not vary much over the opponent's strategy because it uses the opponent's offers to determine the size of a concession but does not try to match the opponent's preferences.

2.8.4 DANS analysis results

In summary the DANS analysis results are as follows. The opponent sensitivity analysis shows a direct link between the correctness and/or completeness of the domain knowledge and opponent sensitivity. The Trade-Off strategy is very sensitive to an opponent given complete information. In that case, the similarity functions exactly match the opponent's preferences and the weights exactly represent the issue importance factors of the opponent and the sensitivity metric for the behaviour of the opponent is converging to zero. Intuitively, the Trade-Off strategy would be more efficient in a smaller domain due to the smaller search space. However, the incomplete domain knowledge in the 2nd Hand Car Selling domain does not allow the Trade-Off strategy to fully use its potential to search for efficient outcomes. The experiments show that if less domain knowledge is available, Trade-Off makes more unfortunate moves.

In general, when issues are predictable, the chance of making an unfortunate step becomes small. This aspect becomes clear in the car domain, where the seller's preferences (Role A) are rather predictable, but the buyer's preferences (Role B) vary a lot.

We conclude that it is impossible to avoid unfortunate moves without sufficient domain knowledge or opponent knowledge. Indeed, the similarity criteria functions used in the Trade-Off Strategy provide general information about the negotiation problem, but do not

take into account the specific attributes of the negotiating parties. In any particular case, a negotiator may deviate from the generalized domain model in various ways.

On the other hand the Bayesian strategy does not use domain knowledge and tries to learn the opponent's preference during negotiation. The learning algorithm of the Bayesian strategy allows it to remain sensitive to the opponent's behaviour and preferences regardless of the completeness and correctness of the available domain knowledge.

Sensitivity to the opponent's preference of the ABMP strategy does not seem to be influenced by the opponent's strategy unlike the Trade-Off strategy. The ABMP strategy shows rather robust performance but it is often outperformed by the smarter strategies, such as the Trade-Off and the Bayesian strategies, in terms of outcome utilities. The Bayesian strategy similar to the ABMP strategy show somewhat more robust behaviour than the Trade-Off strategy. This can be explained by the way it learns the opponent's preferences. It does not require that the opponent's offers stay close to the Pareto frontier. A better model of the opponent's preferences allows the Bayesian strategy to be more sensitive to the opponent's preferences and, finally, reach better negotiation outcomes.

2.9 Conclusions

This paper shows that an analysis of the negotiation dance [15] is important for the understanding and improvement of negotiation strategies. The DANS analysis method introduced in this paper focuses on the classification of negotiation moves and a metrics over this classification. The classification enables us to relate the intent of a strategy in making a negotiation step with the actuality of the perception of that step by the opponent. For example, a strategy might be concession oriented, i.e., moves are intended to be concessions, but in reality some of these moves might be unfortunate, meaning that although the proposer of the bid is giving in, from the perception of the receiver, the bid is actually worse than the previous bid.

By testing strategies over various domains and against various opponents patterns emerge of when such unfortunate moves occur. These patterns are related to dynamic properties such as the sensitivity of strategies. Experiments with DANS show, for example, that the Trade-Off strategy is rather responsive to the behaviour of the opponent, in that it follows the behaviour of the opponent. If that is rather wild, such as the random behaviour of the Random Walker, Trade-Off shows a high percentage of unfortunate moves.

Experiments further show that the occurrence of unfortunate moves is related to features of the negotiation domain and the extent to which such features are incorporated in the strategy. The same holds for knowledge about the preferences of the opponent. To better understand the relative importance of each relation, we have emphasized the distinction between domain knowledge and opponent knowledge.

We think it is impossible to avoid unfortunate moves without sufficient domain knowledge or opponent knowledge. Domain knowledge provides generalized information about the

negotiation problem, but does not necessarily match with individual preferences of negotiating parties. Opponent knowledge concerns individual information and as such is not transferable to other opponents. Therefore, we advocate a combination of domain and opponent knowledge.

The DANS analysis method focuses explicitly on properties of interest to the researcher. The combination of statistical methods and graphical representation is strong: Humans process graphs faster than tables with numbers, however, the number of experiments typically done make it impossible for the human to view every graph produced by the experiment. A more general aspect is that graphs of long negotiation dances become hard to grasp; what may look like a neat series of nice moves, might actually be a mixture of unfortunate and nice moves.

The example in the paper shows how the focus on the percentage of unfortunate moves makes it possible for the DANS method to present to us with particularly insightful graphs, such as the graph of the Trade-Off vs Random Walker that provides insight into the sensitivity of the Trade-Off strategy. Furthermore, based on the sensitivity analyses we showed that the Bayesian strategy is able to overcome the inefficiency of the Trade-Off strategy in domains for which the available domain knowledge is incomplete and/or incorrect.

Future work We believe that our results also show the need for benchmark problems for bilateral negotiation. An interesting direction for future research in this area would be to propose a measure for exploitability of a negotiation strategy. A good negotiation strategy must be able to withstand an inefficient opponent strategy, such as Random Walker, and a strategy that tries to exploit its opponent.

In this paper we showed that the negotiation domain can have strong influence on performance of a negotiation strategy. Thus, another interesting direction for the future work is to develop a design method to generate varying negotiation domains and preference profiles and add this domain- and profile generation method to DANS. While it is impossible to test a negotiation strategy in all possible negotiation scenarios the method should be able to give a good spread of negotiation domains, preference profiles and negotiation strategies such that it covers factors such as the size of the negotiation domain, predictability of the preferences, opposition of preferences, and opponent behaviour (i.e., strategy).

Bibliography

- [1] T. Bosse and C.M. Jonker. Human vs. computer behaviour in multi-issue negotiation. In T. Ito, H. Hattori, T. Matsuo, and M. Zhang, editors, *Proceedings of the First International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems, RRS'05*, IEEE Computer Society Press, pages 11–24, 2005.
- [2] T. Bosse, C.M. Jonker, and J. Treur. Experiments in human multi-issue negotiation:

- Analysis and support. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'04*, page 672–679. IEEE Computer Society Press, 2004.
- [3] R.M. Coehoorn and N.R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings of the 6th International Conference on E-Commerce*, pages 59–68, 2004.
 - [4] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
 - [5] Dhananjay K. Gode and Shyam Sunder. Allocative efficiency in markets with zero intelligence (zi) traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.
 - [6] P. Henderson, S. Crouch, R.J. Walters, and Q. Ni. Comparison of some negotiation algorithms using a tournament-based approach. *Agent Technologies, Infrastructure, Tools and Applications for E-Services*, pages 137–150, 2003. Springer-Verlag.
 - [7] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Towards an open negotiation architecture for heterogeneous agents. In *Proceedings of the 12th international workshop on Cooperative Information Agents XI (CIA 2008)*. Springer.
 - [8] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Negotiation dynamics: Analysis, concession tactics, and outcomes. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*, pages 427–433, 2007.
 - [9] Koen Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the AAMAS 2008*, 2008.
 - [10] Koen Hindriks and Dmytro Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. In *Proceedings of 12th Int. Workshop on Cooperative Information Agents*. Springer-Verlag, 2008.
 - [11] Catholijn M. Jonker and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems Journal*, 15:221–252, 2001.
 - [12] R. Lin, S. Kraus, J. Wilkenfeld, and J. Barry. An automated agent for bilateral negotiation with bounded rational agents with incomplete information. In *Proc. of the 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 270–274, 2006.
 - [13] A.R. Lomuscio, M. Wooldridge, and N.R. Jennings. A classification scheme for negotiation in electronic commerce. *Int. Journal of Group Decision and Negotiation*, 12(1):31–56, 2000.

- [14] C. Mudgal and J. Vassileva. Bilateral negotiation with incomplete and uncertain information. In *Proc. of the Workshop on Cooperative Information Agents*, pages 107–118. Springer, 2000.
- [15] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.
- [16] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2003.
- [17] J.S. Rosenschein and G. Zlotkin. *Rules of encounters Designing conventions for automated negotiation among computers*. Cambridge, MA: MIT Press, 1994.

Negotiation Domain	Strategy of Agent A	Metric	Strategy of Agent B															
			ABMP								Trade-Off							
			Number of steps				Sensitivity				Number of steps				Sensitivity			
			Fort	Nice	Cons	Silent	Selfish	Unfort.	Behav.	Opp.Prefs	Fort	Nice	Cons	Silent	Selfish	Unfort.	Behav.	Opp.Prefs
Car domain	ABMP	Agent B	7%	6%	67%	6%	0%	14%	3.98	0.26	0%	23%	47%	12%	0%	18%	2.34	0.28
		Agent A	1%	3%	54%	20%	0%	21%	1.41	0.24	1%	12%	42%	25%	0%	21%	1.19	0.20
	Trade-Off	Agent B	5%	16%	60%	6%	0%	13%	4.23	0.25	3%	14%	53%	18%	0%	13%	2.24	0.05
		Agent A	0%	10%	45%	20%	0%	25%	1.23	0.12	2%	4%	63%	23%	0%	7%	2.27	0.04
	Random Walk	Agent B	1%	1%	81%	8%	2%	8%	4.63	0.23	1%	29%	22%	18%	1%	30%	1.07	0.31
		Agent A	29%	3%	22%	8%	17%	22%	1.18	0.45	28%	9%	23%	2%	17%	21%	1.47	0.49
	Bayesian	Agent B	4%	12%	66%	6%	0%	12%	4.32	0.25	4%	10%	58%	19%	0%	9%	2.57	0.02
		Agent A	10%	43%	39%	3%	0%	5%	11.50	0.03	3%	12%	62%	20%	0%	3%	3.35	0.01
	Average	Agent B	4%	9%	68%	6%	1%	12%	4.29	0.25	2%	19%	45%	17%	0%	17%	2.06	0.17
		Agent A	1%	6%	49%	21%	2%	21%	1.32	0.22	2%	13%	46%	21%	0%	18%	1.90	0.15
Service Oriented Negotiation	ABMP	Agent B	0%	0%	91%	9%	0%	0%	10.55	0.39	0%	6%	62%	5%	0%	26%	2.19	0.16
		Agent A	0%	2%	83%	16%	0%	0%	5.40	0.39	0%	4%	67%	28%	0%	0%	2.55	0.48
	Trade-Off	Agent B	0%	3%	67%	29%	0%	0%	2.42	0.35	0%	0%	81%	1%	0%	18%	4.31	0.09
		Agent A	0%	0%	77%	6%	0%	17%	3.30	0.20	0%	5%	85%	3%	0%	7%	9.02	0.01
	Random Walk	Agent B	6%	1%	86%	6%	1%	0%	11.63	0.38	2%	39%	20%	3%	1%	35%	1.57	0.13
		Agent A	20%	2%	26%	8%	24%	21%	0.91	0.53	17%	4%	31%	3%	26%	18%	1.08	0.42
	Bayesian	Agent B	0%	1%	87%	4%	0%	8%	7.33	0.38	0%	8%	80%	3%	0%	9%	7.33	0.01
		Agent A	0%	3%	84%	4%	0%	9%	6.69	0.01	0%	9%	82%	3%	0%	3%	15.17	0.01
	Average	Agent B	1%	1%	83%	12%	0%	2%	7.98	0.38	0%	13%	61%	3%	0%	22%	3.85	0.10
		Agent A	1%	3%	81%	15%	1%	2%	6.14	0.42	1%	9%	68%	6%	0%	16%	4.90	0.13
AMPO vs City	ABMP	Agent B	0%	0%	65%	35%	0%	0%	1.89	0.44	0%	0%	56%	12%	0%	32%	1.27	0.36
		Agent A	0%	12%	38%	49%	0%	0%	1.02	0.43	0%	4%	64%	29%	0%	3%	2.13	0.40
	Trade-Off	Agent B	0%	4%	75%	21%	0%	0%	3.78	0.40	0%	6%	80%	5%	0%	9%	6.14	0.03
		Agent A	0%	0%	68%	5%	0%	27%	2.17	0.35	0%	8%	76%	6%	0%	10%	5.25	0.01
	Random Walk	Agent B	0%	0%	59%	37%	1%	3%	1.47	0.46	1%	26%	30%	6%	0%	37%	1.33	0.45
		Agent A	21%	4%	25%	3%	25%	23%	0.98	0.82	22%	3%	26%	4%	22%	23%	1.04	0.82
	Bayesian	Agent B	0%	3%	79%	18%	0%	0%	4.56	0.39	0%	0%	82%	6%	0%	6%	6.83	0.04
		Agent A	12%	45%	34%	5%	0%	5%	9.10	0.04	10%	48%	35%	4%	0%	3%	13.29	0.03
	Average	Agent B	0%	2%	70%	28%	0%	1%	2.93	0.42	0%	8%	62%	7%	0%	21%	3.89	0.22
		Agent A	1%	7%	52%	38%	0%	2%	1.56	0.42	1%	8%	63%	10%	0%	19%	4.07	0.21

Negotiation Domain	Strategy of Agent A	Metric	Strategy of Agent B															
			ABMP								Trade-Off							
			Number of steps				Sensitivity				Number of steps				Sensitivity			
			Fort	Nice	Cons	Silent	Selfish	Unfort.	Behav.	Opp.Prefs	Fort	Nice	Cons	Silent	Selfish	Unfort.	Behav.	Opp.Prefs
Car domain	ABMP	Agent B	7%	6%	67%	6%	0%	14%	3.98	0.26	0%	23%	47%	12%	0%	18%	2.34	0.28
		Agent A	1%	3%	54%	20%	0%	21%	1.41	0.24	1%	12%	42%	25%	0%	21%	1.19	0.20
	Trade-Off	Agent B	5%	16%	60%	6%	0%	13%	4.23	0.25	3%	14%	53%	18%	0%	13%	2.24	0.05
		Agent A	0%	10%	45%	20%	0%	25%	1.23	0.12	2%	4%	63%	23%	0%	7%	2.27	0.04
	Random Walk	Agent B	1%	1%	81%	8%	2%	8%	4.63	0.23	1%	29%	22%	18%	1%	30%	1.07	0.31
		Agent A	29%	3%	22%	8%	17%	22%	1.18	0.45	28%	9%	23%	2%	17%	21%	1.47	0.49
	Bayesian	Agent B	4%	12%	66%	6%	0%	12%	4.32	0.25	4%	10%	58%	19%	0%	9%	2.57	0.02
		Agent A	10%	43%	39%	3%	0%	5%	11.50	0.03	3%	12%	62%	20%	0%	3%	3.35	0.01
	Average	Agent B	4%	9%	68%	6%	1%	12%	4.29	0.25	2%	19%	45%	17%	0%	17%	2.06	0.17
		Agent A	1%	6%	49%	21%	2%	21%	1.32	0.22	2%	13%	46%	21%	0%	18%	1.90	0.15
Service Oriented Negotiation	ABMP	Agent B	0%	0%	91%	9%	0%	0%	10.55	0.39	0%	6%	62%	5%	0%	26%	2.19	0.16
		Agent A	0%	2%	83%	16%	0%	0%	5.40	0.39	0%	4%	67%	28%	0%	0%	2.55	0.48
	Trade-Off	Agent B	0%	3%	67%	29%	0%	0%	2.42	0.35	0%	0%	81%	1%	0%	18%	4.31	0.09
		Agent A	0%	0%	77%	6%	0%	17%	3.30	0.20	0%	5%	85%	3%	0%	7%	9.02	0.01
	Random Walk	Agent B	6%	1%	86%	6%	1%	0%	11.63	0.38	2%	39%	20%	3%	1%	35%	1.57	0.13
		Agent A	20%	2%	26%	8%	24%	21%	0.91	0.53	17%	4%	31%	3%	26%	18%	1.08	0.42
	Bayesian	Agent B	0%	1%	87%	4%	0%	8%	7.33	0.38	0%	8%	80%	3%	0%	9%	7.33	0.01
		Agent A	0%	3%	84%	4%	0%	9%	6.69	0.01	0%	9%	82%	3%	0%	3%	15.17	0.01
	Average	Agent B	1%	1%	83%	12%	0%	2%	7.98	0.38	0%	13%	61%	3%	0%	22%	3.85	0.10
		Agent A	1%	3%	81%	15%	1%	2%	6.14	0.42	1%	9%	68%	6%	0%	16%	4.90	0.13
AMPO vs City	ABMP	Agent B	0%	0%	65%	35%	0%	0%	1.89	0.44	0%	0%	56%	12%	0%	32%	1.27	0.36
		Agent A	0%	12%	38%	49%	0%	0%	1.02	0.43	0%	4%	64%	29%	0%	3%	2.13	0.40
	Trade-Off	Agent B	0%	4%	75%	21%	0%	0%	3.78	0.40	0%	6%	80%	5%	0%	9%	6.14	0.03
		Agent A	0%	0%	68%	5%	0%	27%	2.17	0.35	0%	8%	76%	6%	0%	10%	5.25	0.01
	Random Walk	Agent B	0%	0%	59%	37%	1%	3%	1.47	0.46	1%	26%	30%	6%	0%	37%	1.33	0.45
		Agent A	21%	4%	25%	3%	25%	23%	0.98	0.82	22%	3%	26%	4%	22%	23%	1.04	0.82
	Bayesian	Agent B	0%	3%	79%	18%	0%	0%	4.56	0.39	0%	0%	82%	6%	0%	6%	6.83	0.04
		Agent A	12%	45%	34%	5%	0%	5%	9.10	0.04	10%	48%	35%	4%	0%	3%	13.29	0.03
	Average	Agent B	0%	2%	70%	28%	0%	1%	2.93	0.42	0%	8%	62%	7%	0%	21%	3.89	0.22
		Agent A	1%	7%	52%	38%	0%	2%	1.56	0.42	1%	8%	63%	10%	0%	19%	4.07	0.21

Figure 2.5: Results of the Step Wise Analysis.

Chapter 3

A Multi-Agent Environment for Negotiation

by section, subsection

In this chapter we introduce the System for Analysis of Multi-Issue Negotiation (SAMIN). SAMIN offers a negotiation environment that supports and facilitates the setup of various negotiation setups. The environment has been designed to analyse negotiation processes between human negotiators, between human and software agents, and between software agents. It offers a range of different agents, different domains, and other options useful to define a negotiation setup. The environment has been used to test and evaluate a range of negotiation strategies in various domains playing against other negotiating agents as well as humans. We discuss some of the results obtained by means of these experiments.

3.1 Introduction

Research on negotiation is done in various research disciplines; business management, economics, psychology, and artificial intelligence. The foundations of negotiation theory are decision analysis, behavioral decision making, game theory, and negotiation analysis. The boost of literature on negotiating agents and strategies of recent years is in line with the continuous advance of ecommerce applications, such as eBay, and Marketplace in which negotiations play a role. In essence it focuses on the development of ever more clever negotiation agents, that are typically tested in one domain, against one or two other negotiation agents, almost never against humans. In our opinion, in order to become acceptable as negotiators on behalf of human stakeholders, negotiation agents will have to prove their worth in various domains, against various negotiation strategies and against human negotiators. In order to gain a better understanding of the negotiation dynamics and the factors that influence the negotiation process it is crucial to not only mathematically evaluate the efficiency of negotiation outcomes but also to look at the pat-

tern of offer exchanges, what Raiffa [30] calls the negotiation dance. In the remainder we present architecture of a formal toolbox to simulate negotiations and analyze patterns in offer exchanges and present some initial findings in the literature. The System for Analysis of Multi-Issue Negotiation¹ (SAMIN) is developed as a research tool, to improve the quality of negotiating agents, and as a training environment to develop negotiation skills of human negotiators. To that purpose SAMIN offers a range of analytical tools, a tournament tool, a preference elicitation tool, and a number of negotiation domains, negotiation agents, and user interfaces for human negotiators.

3.2 Application Domain

Negotiation is an interpersonal decision-making process necessary whenever we cannot achieve our objectives single-handedly [32]. Pruitt [28] emphasizes the process of negotiation and the fact that the outcome should be a joint decision by the parties involved. Typically each party starts a negotiation by offering the most preferred solution from the individual area of interest. If an offer is not acceptable by the other parties they make counter-offers in order to move each other closer to an agreement. The field of negotiation can be split into different types, e.g. along the following lines: (a) one-to-one versus more than two parties; (b) single- versus multi-issues; (c) closed versus open (d) mediator-based versus mediator-free. The research reported in this chapter concerns one-to-one, multi-issue, closed, mediator-free negotiation. A special case of one-to-many negotiation is considered. In this case, an auction mechanism [10] is approximated by a negotiation setup [16]. For more information on negotiations between more than two parties (e.g., in auctions), the reader is referred to, e.g., [31]. In single-issue negotiation, the negotiation focuses on one aspect only (typically price) of the object under negotiation. Multi-issue negotiation (also called multi-attribute negotiation) is often seen as a more cooperative form of negotiation, since often an outcome exists that brings joint gains for both parties, see [30]. Closed negotiation means that no information regarding preferences is exchanged between the negotiators. The only information exchanged is formed by the bids. More information about (partially) open negotiations can be found, e.g., in [20] and [30]. However, the trust necessary for (partially) open negotiations is not always available. The use of mediators is a well-recognised tool to help the involved parties in their negotiations, see e.g., [19, 30]. The mediator tries to find a deal that is fair to all parties. Reasons for negotiating without a mediator can be the lack of a trusted mediator, the costs of a mediator, and the hope of doing better. The SAMIN system is developed to support research into the analysis of negotiation strategies. The analysis of negotiation strategies provides new insights into the development of better negotiation strategies.

Negotiation parties need each other to obtain an outcome which is beneficial to both and is an improvement over the current state of affairs for either party. Both parties need to

¹This negotiation environment, user manuals, and a number of implemented negotiation agents can be downloaded from <http://mmi.tudelft.nl/negotiation>.

believe this is the case before they will engage in a negotiation. Although by engaging in a negotiation one party signals to the other party that there is potential for such gain on its side, it may still leave the other party with little more knowledge than that this is so. Research shows that the more one knows about the other party the more effective the exchange of information and offers [30]. Furthermore, humans usually do have some understanding of the domain of negotiation to guide their actions, and, as has been argued, a machine provided with domain knowledge may also benefit from such domain knowledge [6]. It is well-known that many factors influence the performance and outcome of humans in a negotiation, ranging from the general mindset towards negotiation to particular emotions and perception of fairness. As emphasized in socio-psychological and business management literature on negotiation, viewing negotiation as a joint problem-solving task is a more productive mindset than viewing negotiation as a competition in which one party wins and the other loses [7, 30, 32]. Whereas the latter mindset typically induces hard-bargaining tactics and rules out disclosure of relevant information to an opponent, the former leads to joint exploration of possible agreements and induces both parties to team up and search for trade-offs to find a win-win outcome. Different mindsets lead to different negotiation strategies. A similar distinction between hard- and soft-bargaining tactics has also been discussed in the automated negotiation system literature where the distinction has been referred to as either a *boulware* or a *conceder* tactics [5]. Emotions and perception of fairness may also determine the outcome of a negotiation. People may have strong feelings about the “rightness” of a proposed agreement. Such feelings may not always be productive to reach a jointly beneficial and efficient agreement. It has been suggested in the literature to take such emotions into account but at the same time to try to control them during negotiation and rationally assess the benefits of any proposals on the table [7, 32]. Apart from the factors mentioned above that influence the dynamics of negotiation, many other psychological biases have been identified in the literature that influence the outcome of a negotiation, including among others partisan perceptions, overconfidence, endowment effects, reactive devaluation [25, 32].

3.2.1 The Added Value of the MAS Paradigm

Negotiation involves conflicting interests, hidden goals, and making educated guesses about the preferences and goals of the other parties involved. A system that supports closed negotiation needs to protect the integrity of the parties or stakeholders that participate in a negotiation and it is natural to provide every stakeholder with an agent of their own. It thus is natural to use the MAS paradigm to model the interaction between negotiating parties. Parties in a negotiation are autonomous and need to decide on the moves to make during a negotiation. This decision problem is particularly complex in a closed negotiation where negotiating parties do not reveal their preferences to each other. Moreover, other factors such as the complexity of the domain of negotiation may pose additional problems that need to be solved by a negotiating agent.

SAMIN contributes to the MAS paradigm as a research tool that facilitates research into

the design of efficient negotiation strategies. The tool more specifically facilitates the evaluation of the performance of a negotiation strategy by means of simulating multiple negotiation sessions and feeding the results of the simulation to the analytical toolbox of SAMIN. We have found that the results of a well-defined negotiation setup may help analysing the strengths and weaknesses of a strategy and may be used to improve a negotiation strategy significantly. It has also been shown that strategies may perform quite differently on different domains. A variety of negotiation domains and agents is available in SAMIN to evaluate a negotiation strategy in different negotiation setups. The open architecture of SAMIN, moreover, facilitates the integration of new negotiation domains and agents.

3.2.2 Design Methods Used

An earlier version of SAMIN, see [2, 18], was designed using the DESIRE method [3]. Redesign was necessary to open the system for agents designed and implemented by others and to ease the definition of new negotiation domains. The redesigned version is implemented in the Java programming language that is supported by the majority of computer platforms.

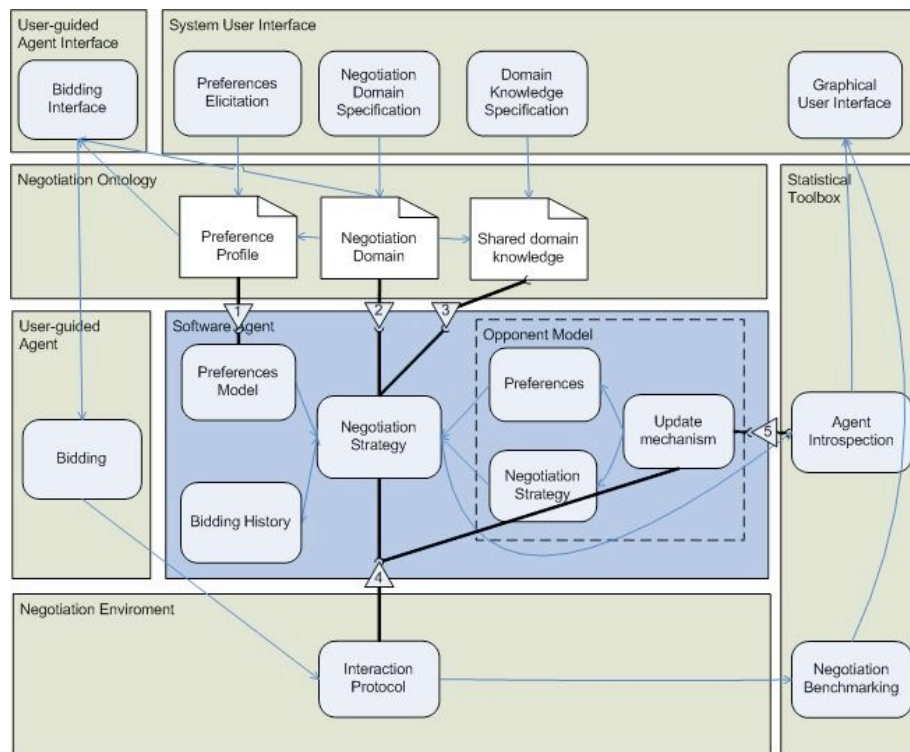


Figure 3.1: The Open Negotiation System Architecture

The current version of SAMIN implements the architecture proposed in [13]. Figure 3.1 illustrates this architecture. The architecture is based on an analysis of the tasks that

need to be supported by a generic negotiation environment that is capable of integrating a variety of negotiating agents and is able to simulate negotiations between such agents. The architecture provides a minimal but sufficient framework including all features necessary to simulate a wide range of negotiation scenarios and to enable integration of negotiation agents. The architecture consists of four main layers, a human bidding interface, and a negotiating agent architecture. The four layers include an *interaction layer*, an *ontology layer*, a *graphical user interface layer*, and an *analytical toolbox*.

The *interaction layer* provides functionality to define negotiation protocols and enables communication between agents (see Section 3.4.2 for details). The *ontology layer* provides functionality to define, specify and store a negotiation domain, and the preferences of the negotiating agents (see Section 3.4.3 for details). The architecture can also be used for education purposes and for the training of humans in negotiation. For that purpose, a *graphical user interface layer* is available that facilitates human user(s) to participate in a negotiation setup (see Section 3.2.3 for details). The *analytical toolbox* provides functionality to organize tournaments between agents, and to review the performance and benchmark results of agents that conducted a negotiation. It provides a variety of tools to analyze the performance of agents and may also be used to compute quality measures related to e.g. the quality of an opponent model [15].

The architecture that is introduced here identifies the main integration points where adapters are needed to connect a negotiating agent to this architecture. The agent architecture itself defines the common components of a negotiating agent. This architecture may be instantiated with various software agents, as illustrated below.

The integration points or interfaces to connect software agents to the negotiation environment which allows them to interact with other agents available in the environment are numbered 1 through 5 in Figure 3.1. To integrate heterogeneous negotiation agents, such agents have to be aligned with these integration points. Alignment by complete redesign of the agent typically requires significant programming efforts and may also cause backward compatibility problems. To minimize the programming efforts, a better approach is to use a set of adapters or wrappers which are used to wrap the agent code. We have used the adapter design pattern [22] for this purpose. The minimal set of adapters that has to be implemented includes a negotiation domain adapter, a preference profile adapter and an interaction protocol adapter, which each correspond to an essential element of a negotiation. The shared domain knowledge adapter and the agent introspection adapter are optional. The shared domain knowledge adapter provides additional knowledge about the domain to all agents, making this knowledge shared and publicly available. The agent introspection adapter facilitates the introspection of internal components of an agent, such as an opponent model. The latter adapter is mainly available for analysis purposes and research. For more details about the adapters the reader is referred to [13].

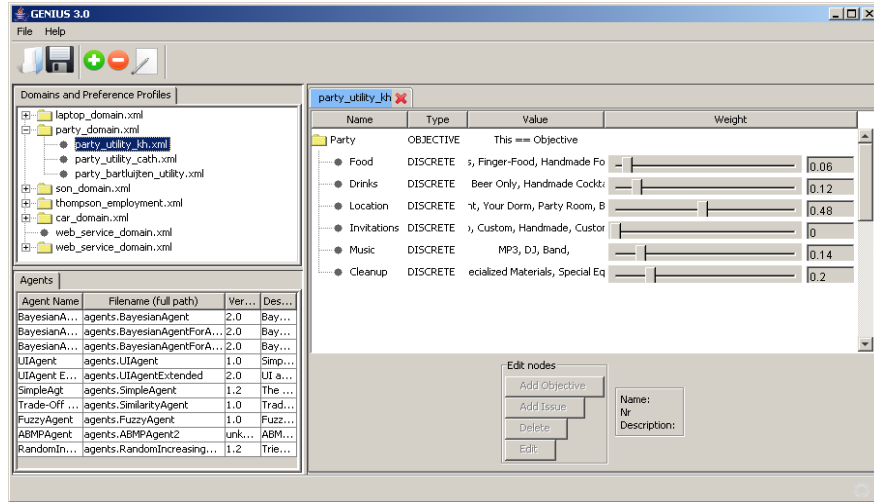


Figure 3.2: A graphical user interface for preferences elicitation.

3.2.3 User Interaction

The user interaction in SAMIN takes place in the graphical user interface layer and can be divided in two categories of user: researchers and human subjects in experiments. We implemented a graphical user interface that enables a user to define the *negotiation game*, the parameters of the negotiation, the subject or domain of negotiation, and preferences of the agents (which also means that the preferences of a human subject can be predefined).

Negotiation Domain and Preference Profile Editor

The *Negotiation Domain and Preference Profile Editor* of SAMIN (see Figure 3.2) is used to create and modify negotiation domains and preference profiles. A *negotiation domain* is a specification of the objectives and issues to be resolved by means of negotiation. An objective may branch into sub-objectives and issues providing a tree-like structure to the domain. The leafs of such a tree representing the domain of negotiation must be the issues that need to be agreed upon in a negotiation. Various types of issues are allowed, including discrete enumerated value sets, integer-valued sets, real-valued sets, as well as a special type of issue used to represent a price associated with the negotiation object. For every issue the user can associate a range of values with a short description and a cost.

A *preference profile* specifies personal preferences regarding possible outcomes of a negotiation. The profile is used to convert any offer in that domain to a value indicating how the user would rate that offer, the so called utility value. The current version of SAMIN supports linear additive utility functions [30]. The profile is also called a *utility space*.

A *weight* that is assigned to every issue indicates the importance of that issue. A human user (see Figure 3.2) can move sliders to change the weights or enter their values by hand, which are automatically normalized by the editor. In the issue editor the user can assign an

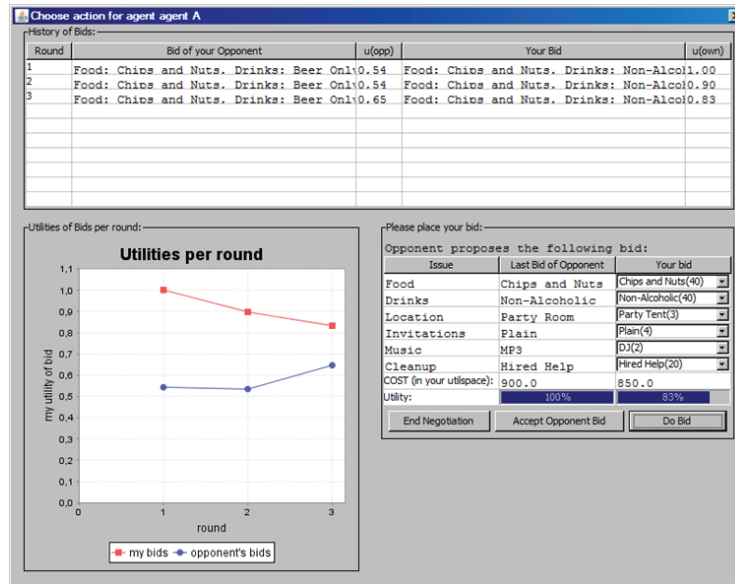


Figure 3.3: Human negotiator graphical user interface.

evaluation to every value of the issue. The evaluation values are positive integers starting with 1. The evaluation values are automatically normalized for each issue to ensure they are in the range $[0; 1]$.

Human Negotiator User Interface

A human subject playing in a negotiation game, is provided with a graphical interface for the bidding phase of the game. The bidding interface is implemented with a dummy agent that exchanges the messages between the graphical user interface (GUI) and the environment. Therefore, the GUI for the human negotiator is not hard coded in SAMIN. The GUI can be easily extended without modifications of the SAMIN code. Furthermore, the dummy agent can be replaced with an algorithm that would provide negotiation a support to the human negotiator. It provides, for example, an analysis of the opponent's behaviour or even advise the human negotiator upon the next offer to propose and an action to be taken.

Figure 3.3 presents human player GUI that is currently available in SAMIN. This GUI has three main components: a bidding history table (top), a utility history plot (bottom left), and a bidding interface (bottom right). The bidding history shows all bids exchanged between the negotiating parties in a single session. The bids are represented by the values assigned to every issue in the negotiation domain. In addition, the utilities of the bids according to the human player's preference profiles are shown in the table. Note that in a closed negotiation the negotiating parties have no access to the preference profiles of each other and, therefore, utilities can be calculated only on the basis of own preference profile.

The bidding interface has two main components: a table showing the last bid and a possible next bid and a row of buttons representing possible actions for the humans negotiator's. The table has three columns:

- the left column shows the names of the issues in the domain;
- the center column shows the values for the issues as proposed in the last bid of the opponent;
- the right column shows the current selected values for the issues. A user can edit the current bid by clicking on the fields, which will open the drop-down boxes in the fields.

The last two rows of the table show the cost and utility of the last opponent's bid and your current bid. The cost field will turn red if the bid exceeds the maximum cost. The utility is shown as a percentage and also as a bar of matching size. These values are computed according to the user's utility space because a user has no access to the opponent's utility space. The lower three buttons allow a user to submit the next bid as it is set in the right column, or to accept the opponent's last bid.

3.3 Agents

In this section we present an agent architecture in SAMIN and explain the state-of-the-art negotiation agents that are available in SAMIN.

3.3.1 Agent Architecture

The *software agent* component highlighted with the darker area in Figure 3.1 is a generic component that can be instantiated by a variety heterogeneous software agents. The components that are specified as part of a software agent in Figure 3.1 are the parts of the *conceptual design* of such agents but do not need to be actually present or identifiable as such in any particular software agent. These components are not introduced here to specify a requirements that need to be satisfied when developing an agent (although it could be used as such [1, 17, 21]). Here these components are introduced to identify integration points of agents with the system architecture. Five of such integration points, also referred to as *adapters*, were identified above.

In the reminder of this section we discuss every component of the proposed agent architecture.

Preference Model The component models the agent's preferences with respect to the set of possible negotiation outcomes. The model can be based on various structures: utility functions, rankings, etc. This component can require additional processing depending

on the complexity of the agent's preferences and the types of inquiries that can be made by other components, see e.g. [19]. Typically, the preferences model must be able to evaluate an outcome on a given scale, compare two or more outcomes, give a single or a set of outcomes that satisfy some constraints on the negotiation domain and preferences.

Negotiation Strategy This is the core component of any negotiation agent. It makes decisions about acceptance of the opponent's offer, ending the negotiation, and sending a counter-offer. To propose a counter-offer the negotiation strategy can use various tactics [5]. Depending on the negotiation tactics used in the negotiation strategy the component can use information about the model of the agent's own preferences, the opponent's preferences and strategy (as far as known to or guessed by the agent), and, the previous offers made during the current, or even previous negotiation sessions.

Negotiation History The negotiation history component keeps track of the bids exchanged between the agents in a negotiation. It can also have a history about earlier negotiations, the outcomes, identities of the opponents, and even opponent models. It can be used by the negotiation strategy component as an additional information source to improve its negotiation performance. For example, in repetitive negotiations with the same opponents this information can be used as a priori knowledge about the opponent to shorten the learning time.

Opponent Model In the negotiation games we consider here, the preferences of negotiation parties are private [30]. Efficiency of a negotiation strategy can be significantly improved with information about the preferences of the opponent [33]. In the literature a number of learning techniques have been proposed to learn the opponent's preferences model from the offers exchanged in a single-shot negotiation, see e.g., [34, 13]. In [33] it was shown that a successful negotiation strategy should make use of an opponent model.

Our generic component consists of three main subcomponents: *preferences*, *negotiation strategy*, and *update mechanism*.

The component *Preferences* contains specifications of the preferences of the current and previous negotiation opponents. As the opponent's preferences are typically private, the preference information has a certain degree of uncertainty. Depending on the agent developed on the base of the generic components information about the certainty of the preferences can be maintained or not.

The aim of the model of *opponent's strategy* is to predict negotiation moves that will be made by the opponent. It is important to know for an agent what the next move of the opponent would be. This knowledge can be used in the negotiation strategy to increase the efficiency of the agent's own offers and increase the chance of acceptance of its offer by the opponent.

Models of the opponent's preferences and strategy are typically learned by the agent from the evidence, such as negotiation agreements achieved in the previous negotiations [33], and offers sent by the opponent in multiple sessions of single-shot negotiations [13, 34, 17]. The learning techniques used in the agent can depend on the types of the models chosen to represent the opponent's preferences and strategy.

3.3.2 State of the Art Negotiating Agents

Interfaces and adapters have been developed to make it easy to integrate agents developed by others into SAMIN, see [13]. A number of the state-of-the-art agents have found a place in SAMIN: ABMP [18], Bayesian agent [14], Bayesian Tit-for-Tat [12], FBM [29], Trade-off agent [6], QO agent [24], Random Walker [11]. As they were developed by different teams, their design, architecture, and implementation varies.

Random Walker The Random Walker strategy introduced in [11], also known as Zero Intelligence (ZI) strategy [8], randomly jumps through the negotiation space. It does not use own preferences or a model of opponent's preferences to make an offer. Random Walker accepts the opponent's offer if it has higher utility than the agent's own last offer. The Random Walker strategy can be run with a break-off point to avoid making offers below that utility and, thus, introduces some limited rationality in its behaviour.

It is difficult for the Random Walker strategy to achieve a better agreement than its break-off point as there is only a very low probability that it will be able to find bids close to Pareto frontier. Any efficient negotiation strategy that is capable of learning an opponent model and is able to use it efficiently would be expected to outperform the Random Walker strategy. For this reason, the Random Walker strategy may be used as a "baseline" strategy. In addition, as the Random Walker strategy does not derive its moves from its preference profile but only uses an acceptance strategy to avoid outcomes with a utility below its break-off point, it also provides a good test case to evaluate of robustness of a negotiation strategy.

ABMP Agent The ABMP strategy is a concession-oriented negotiation strategy, see [18]. It selects counter-offers without taking domain or opponent knowledge into account. The ABMP strategy decides on a negotiation move based on considerations derived from the agent's own utility space only. It calculates a utility of a next offer, called *target utility*, based on the current utility gap between the last opponent's offer and the last own offer. To determine the next offer the target utility is propagated to the individual issues taking into account the weights of the issues in the agent's preferences profile. The ABMP strategy can be fine-tuned with a number of parameters, such as the negotiation speed, concession factor, configuration tolerance and others.

The original ABMP strategy was not capable of learning. A heuristic for adapting the ABMP strategy to the opponent's issue priorities was introduced in [17]. The results

showed improvement of the negotiation outcome compared to the original version of the ABMP strategy.

The ABMP strategy was implemented in an ad hoc environment using the DESIRE method [3]. The environment facilitated negotiation about a Second-hand car domain [18] that was hard-coded in the implementation. Later, when the second Java-based version of the SAMIN was available the ABMP strategy was re-implemented in SAMIN. The results of the DESIRE-based ABMP implementation were reproduced in SAMIN.

Trade-off Agent The effectiveness of using knowledge about the negotiation domain has been demonstrated in the Trade-off strategy introduced in [6]. In particular, this paper shows that domain knowledge (coded as so-called similarity functions) can be used to select bids that are close to the opponent's bids, thus increasing the likelihood of acceptance of a proposed bid by that opponent. In this approach, the knowledge represented by similarity functions is assumed to be public.

In [6], the Trade-off strategy is combined with several so called *meta strategies* that control the concession behaviour of the agent. The most interesting meta strategy, the *smart* strategy, consists of deploying a Trade-off mechanism until the agent observes a deadlock in the average closeness of own offers compared to that of the opponent as measured by the similarity function. In a case of the deadlock, the value of the previous offer is reduced by a predetermined amount (0.05), thereby lowering the input value of the Trade-off mechanism.

The Trade-off strategy was originally evaluated on the Service-Oriented Negotiation (SON) domain. The SON domain has four quantitative continuous issues, the price, quality, time, and penalty. Both, buyer and seller use linear functions to evaluate individual issues and combine them in a linear additive utility function using a vector of weights. It is assumed that the buyer and the seller have opposite preferences for every issue, that is, if buyer wants to maximize the quality then the seller wants to minimize it. Therefore, in this domain the differences in the weights are the key elements to consider for joint improvements of the offers.

The Trade-off strategy combined with the smart meta strategy showed good performance on the SON in the experimental setup of [6]. It was demonstrated that the Trade-off strategy is capable of producing very efficient offers resulting in agreements that are very close to the Pareto efficient frontier. Interestingly, the best performance the Trade-off strategy showed in negotiation against itself, while in negotiations against agents that used other meta strategies the utility of agreement was somewhat lower. This phenomenon will be discussed in details in Section 3.6.

Unfortunately, no implementation of the Trade-Off strategy was available. The strategy was implemented in the SAMIN from scratch. The results reported in [6] were reproduced for the Service-Oriented Negotiation domain.

Bayesian Agent One way to approach the problem of incomplete information in closed negotiation is to learn an opponent's preferences given the negotiation moves that an opponent makes during the negotiation. A learning technique based on Bayesian learning algorithm was proposed in [14]. The opponent model in [14] is based on learning probability over a set of hypothesis about evaluation functions and weights of the issues. The probability distribution is defined over the set of hypothesis that represent agent's belief about opponent's preferences. Structural assumptions about the evaluation functions and weights are made to decrease the number of parameters to be learned and simplify the learning task.

The set of hypotheses about the evaluation function is defined using three types of shapes of the functions: (a) downhill shape: minimal issue values are preferred over other issue values, and the evaluation of issue values decreases linearly when the value of the issue increases; (b) uphill shape: maximal issue values are preferred over other issue values, and the evaluation of issue values increases linearly when the value of the issue increases; (c) triangular shape: a specific issue value somewhere in the issue range is valued most and evaluations associated with issues to the left ("smaller") and right ("bigger") of this issue value linearly decrease (think, e.g., of an amount of goods).

During a negotiation every time when a new bid is received from the opponent the probability of each hypothesis is updated using Bayes' rule. This requires a conditional probability that represents the probability that the bid might have been proposed given a hypothesis. Therefore the utility of bid is calculated according to this hypothesis and compared with the predicted utility according to the rationality assumption. To estimate the predicted utility value an assumption about the opponent concession tactics is used based on a linear function.

Authors propose two versions of the learning algorithm. In the first version of the algorithm each hypotheses represents a complete utility space as a combination of weights ranking and shapes of the issue evaluation functions. The size of the hypothesis space growth exponentially with respect to the number of issue and thus is intractable for negotiation domains with high number of issues.

The second version of the algorithm is a scalable variant for the first one. This version of the agent tries to learn probability distribution over the individual hypothesis about the value of the weight and shape of the issue evaluation function independently of other issues. The computational tractability of the learning is achieved by approximating the conditional distributions of the hypotheses using the expected values of the dependent hypotheses.

QO Agent In [24] the authors propose a negotiation agent, called QO agent, that is based on qualitative decision making. The QO agent is designed for automated negotiations with multiple issues. The internal structure of the QO agent is similar to the agent architecture proposed in this article. The underlying assumption in the QO agent is that the opponent uses one of three preference profiles. The preference profiles of the oppo-

nent are represented in same way as QO agent's own preference profile. A probability is associated with each of the possible opponent profiles. An update mechanism interprets the observed offers from the opponent and updates the probability distribution according to the opponent strategy model. The opponent profiles have the same structure as the own preferences profile and the same preference profile adapter is used to load them from files.

The original implementation of the QO agent uses Java programming language. The interaction protocol, however, is more complex than the alternating offers protocol currently used by the SAMIN. The QO agent environment implements a rather complex interaction protocol that extends the alternating offers protocol. It does not have a clear turn taking flow and allows agents to exchange pre-defined textual messages between the agents, such as threats of breaking negotiation if the last offer is not accepted. It was decided to simplify it in the interaction protocol adapter. Only those functions of the agent were used that represent the core functionality: interpret the opponent's offer, generate next action of the agent, generate a counter-offer.

Fuzzy-based Model Agent The other agent integrated into the negotiation system is the Fuzzy-based model (FBM) agent introduced in [29]. The Fuzzy-based agent is designed for negotiation where agents can exchange fuzzy proposals. The original FBM agent is designed for negotiations where agents can exchange fuzzy proposals. The original implementation of the FBM agent works only for one-issue negotiations but can be extended for multi-issue negotiations. As a result, the negotiation domain is defined using one issue that takes real values from a give interval. The agent adopts time dependent negotiation tactics from [5] and, thus, always makes concession towards opponent. The offers are defined using two values: the peak value and the stretch of the offer.

The FBM agent is implemented in an experimental setup using Java programming language. The experimental setup uses the alternating offers protocol [27]. The preference profile is hard-coded in the agent and based on a linear function. The experimental setup consists two agents that have opposed preferences over the issues.

Bayesian Tit-for-Tat Agent In [12] a negotiation strategy is proposed that uses a model of the opponent's preferences not only to increase the efficiency of the negotiated agreement but also to avoid exploitation by the other party in a sophisticated Tit-for-Tat manner. Authors in [12] try to show that two important goals in any negotiation can be realized when a reasonable estimate of the preferences of an opponent is available.

For that purpose they combine the Bayesian learning technique as proposed in [14] with a Tit-for-Tat tactic, see e.g., [5], and the classification of negotiation moves as described in, e.g., [11]. As is typical for Tit-for-Tat, it avoids exploitation by a form of mirroring of the bids of the opponent. Bayesian learning is used to learn the opponent's preferences. The opponent profile together with the classification scheme is used to develop a sophisticated Tit-for-Tat Bayesian negotiation strategy.

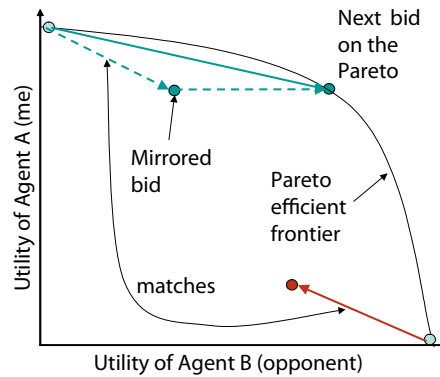


Figure 3.4: Bayesian Tit-for-Tat Strategy

Bidding of the proposed strategy can be understood by the opponent as signalling whether a move is appreciated or not (which is not as easy as it seems). Tit-for-Tat Bayesian negotiation strategy does not punish the opponent for making a move that can be understood as an honest mistake. The strategy is based on a rationality assumption, i.e., that an opponent would tend to accept more preferred offers over less preferred. In line with this assumption the strategy searches for Pareto efficient offers, i.e., offers that cannot be improved for both parties simultaneously. Pareto efficient offers increase the chances that an opponent accepts an offer, while protecting the agent's own preferences as best as possible. Finding such offers requires that the Pareto efficient frontier can be approximated which is only feasible if a reasonable model of the opponent's preferences is available.

The basic idea of Tit-for-Tat in multi-issue negotiation is to respond to an opponent move with a symmetrical one, as depicted in Figure 3.4. Typically, a rational negotiation strategy would try to make concession moves at some points during the negotiation. The most reasonable response to a concession move would be a concession move of approximately the same concession size. This is called "mirroring" the move of the opponent.

Mirroring simply in this manner would imply that an unfortunate move (an offer that decreases utility for both parties compared to the agent's previous offer) of the opponent would be answered with an unfortunate step. However, it is not rational to consciously make unfortunate steps. Therefore, authors conclude that the pure tactic by mirroring the opponent moves is too simplistic. Instead they use an approximation of the Pareto frontier computed using the learned opponent model and the agent's own preference profile to add an additional step.

The Bayesian Tit-for-Tat strategy is constructed on the basis of the assumption that by maximizing the opponent's utility in every offer, the chance of acceptance increases as well. Therefore, if after mirroring the opponent's move the efficiency of the agent's own next move can be increased by selecting an equivalent offer (with respect to the agent's preference profile) on the Pareto frontier the strategy will choose to make that offer. Important is that this approach makes the Bayesian Tit-for-Tat negotiation strategy less dependent on the efficiency of the opponent's strategy. The opponent might intend to make

a concession but in fact make an unfortunate move. By selecting a bid on the approximated Pareto frontier, while mirroring the concession intent of the opponent, the strategy is able to maintain a high efficiency of the outcome, no matter what mistakes the opponent makes.

3.4 Multi-Agent System

The organisation of SAMIN as a multi-agent system and as research environment is introduced in [13].

3.4.1 Organisation

Negotiation, in fact, can take place in a distributed environment. To support distributed negotiation a Web-based interface to the system will be introduced in the next version. This will enable negotiations between humans that are physically distributed. In addition, the Web interface will allow researchers to upload their code from different locations and participate in a tournament.

To setup a negotiation a negotiation template is created. Negotiation template specifies all details of the negotiation: number of agents (currently only bilateral negotiations are supported), names of the agent's classes that implement negotiation strategies, negotiation domain and preference profiles of the parties. This setup is static through single negotiation session.

The structure of the multi-agent system and organisation of the negotiating agents in SAMIN is determined by the negotiation protocol that is used. The interaction of agents is also fully controlled by the environment and negotiation protocol used. All agents are required to comply with the protocol, which is enforced by the environment.

3.4.2 Interaction

The interaction layer manages the rules of encounter or protocol that regulate the agent interaction in a negotiation. Any agent that wants to participate in such a negotiation protocol must accept and agree to conform to these rules. An interaction protocol specifies which negotiation moves and what information exchange between agents is allowed during a negotiation.

The current version of SAMIN focuses on bilateral negotiation. A centralized interaction engine is used, which facilitates the control over the negotiation flow and the enforcement of rules on the negotiation process. The interaction engine also feeds information to the advanced logging capabilities of SAMIN. Logs are used by the analytical toolbox to assess the performance of negotiation strategies and algorithms, see [11, 13]. Interaction

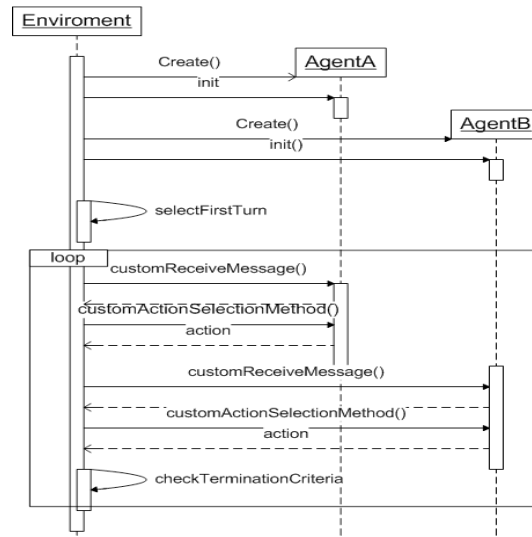


Figure 3.5: A sequence diagram of the interaction protocol

protocols are implemented in the negotiation environment as a separate component to allow the use of a variety of protocols. Implementation of a new interaction protocol in the negotiation environment is a relatively easy task and has no or minimal effect on the agent code.

An example of one of the best known negotiation protocols, the alternating offer protocol [27], is illustrated in Figure 3.5. The alternating offers protocol in a bilateral setting dictates a simple turntaking scheme where each agent is allowed to make a single negotiation move when it is its turn. Apart from turntaking a protocol may also dictate whether exchange of complete package deals is required or that alternatively the exchange of partial bids is allowed. In addition a protocol may manage deadlines, or timeouts that are fixed by the environment.

The interaction protocol is initialized with the information provided by the user. There is no need for a yellow pages mechanism as the agents are made aware about the identity of each other and thus are able to keep track of previous negotiations with the same partner if multiple negotiation sessions are played.

In [16] an alternative protocol involving multiple agents is introduced that is also available in SAMIN. The motivation for introducing this protocol is that it can be used to simulate an auction mechanism. [16] shows that a particular auction mechanism, called the Qualitative Vickrey Auction (QVA) [10], can be simulated with the protocol.²

The QVA mechanism can be thought of as consisting of two rounds. In the first round, the buyer publicly announces her preferences, potential service providers (sellers) submit offers in response, and a winner is selected by the buyer. The winner is the seller who has submitted the best offer from the point of view of the buyer. After establishing the winner,

²The QVA is a generalization of the well-known Vickrey auction to a multi-issue setting where payments are not essential. In QVA a buyer has complex preferences over a set of issues.

in a second round, the buyer determines the second-best offer (from its perspective again) it received from another seller, announces this publicly, and then the winner is allowed to select any agreement that has at least the same utility to the buyer as the second-best offer (which can be determined by the winner since the preferences of the buyer are publicly announced). It is assumed that the bids proposed in the first round are all monitored by a trusted third party.

The negotiation protocol of [16] provides an alternative to the QVA mechanism. An advantage of using a negotiation setup instead of the QVA is that in that case the buyer does not have to publicly announce its preferences. The negotiation protocol is structured in two rounds to match the structure of the mechanism. In the first round negotiation sessions are performed between the buyer and every potential seller using the Alternating offers protocol (see Figure 3.5). Moreover, the negotiation sessions are assumed to be independent. At the end of the first round, a winner (one of the sellers) is determined. Before starting the second round, the agreement between the seller and buyer that is second-best from the perspective of the buyer is revealed to all sellers, in particular to the winner. In the second round an agreement between the winner and the buyer is established. In section 3.6 we present some experimental results received for the proposed negotiation mechanism.

3.4.3 MAS Environment

The MAS environment in SAMIN is a negotiation environment that controls some aspects of the agent's behaviour, such as the setup and initialization of a negotiation session(s), compliance of the agents with a selected negotiation protocol, etc. The layers with corresponding components of the negotiation environment are shown in Figure 3.1 and have a lighter background. First of all, the negotiation environment provides a negotiation ontology to the agents. The ontology specifies concepts, such as a negotiation domain, a preference profile, and shared knowledge.

A *negotiation domain* is a specification of the objectives and issues to be resolved by means of negotiation. It specifies the structure and content of bids or offers exchanged, and of any final outcome or agreement. An outcome determines a specific value for each issue, or, alternatively, only for a subset of the issues. Objectives allow to define a tree-like structure with either other objectives again or issues as children, in line with [30]. Various types of issues are allowed, including discrete enumerated value sets, integer-valued sets, real-valued sets, as well as a special type of issue called *price* issue. Additionally, a specification of a negotiation domain may introduce constraints on acceptable outcomes. For example, costs associated with a particular outcome may not exceed the available budget of the agent.

A *preference profile* specifies the preferences regarding possible outcomes of an agent. It can be thought of as a function mapping outcomes of a negotiation domain onto the level of satisfaction an agent associates with that outcome. The structure of a preference profile

for obvious reasons resembles that of a domain specification. The tree-like structure allows to specify relative priorities of parts of the tree. This allows, for example, to ensure that all issues relating to travelling combined are weighted equally as all issues relating to the actual stay at a particular location.

In a *closed* negotiation an agent is not informed about the preferences of its negotiating partner. In that case an agent can at best use a reconstruction (using e.g. machine learning techniques) of these preferences to decide on the negotiation move it should do next. It is typical, however, that with a domain comes certain public knowledge that is shared and can be used to obtain a better negotiation outcome. For example, common preferences such as preferring early delivery over later (though not always the case) may be common knowledge in a given domain. Such knowledge allows agents to compute the preferences of their negotiation partner e.g. using the time interval between two dates. This type of knowledge, labelled *shared domain knowledge*, is modelled explicitly as a separate component that can be accessed by all negotiating agents.

The *analytical toolbox* layer of the negotiation environment a set of statistical analysis methods to perform an outcome analysis on negotiation sessions as introduced and discussed in e.g., [11, 30]. Furthermore, the toolbox contains methods for the analysis of dynamic properties of negotiation sessions as discussed in e.g., [11]. The methods for both outcome and dynamics analysis were used to produce a number of performance benchmarks for negotiation behaviour and for the agent components [13]. The analytical toolbox uses the optimal solutions [30], such as the Pareto efficient frontier, Nash product and Kalai-Smorodinsky solution for the negotiation outcome benchmarking. The benchmarks in the negotiation system can be used to analyze the performance of opponent modelling techniques, the efficiency of negotiation strategies, and the negotiation behaviour of the agent. The result of the analysis can help researchers to improve their agents. The output of the analytical toolbox is presented graphically (see e.g., Figures 3.6 and 3.8).

3.5 Execution Platform

The system is implemented as a stand-alone application running on a single computer. The negotiation settings, such as role and types of the agents, negotiation domain, and preference profiles are predefined by an script. A tournament is a typical experimental setup for negotiating agents [11]. Therefore, the system has a utility to generate scripts for a tournament setup and can automatically run a sequence of negotiation.

SAMIN is currently focused on the closed negotiations, where negotiating parties have no access to the preference profiles of each other. In addition, agent's own preference profile is supposed to be static during negotiation and cannot be changed during the negotiation. Few security precautions were implemented in SAMIN to meet these requirements and avoid situations where agents would improve their performance by means of software

hacks. This is especially important when SAMIN is used as a testbed for negotiating agents or in an educational setup.

Negotiating agents in SAMIN as any imperfect software product can fail. All errors and exception raised by the agent's code are properly logged by the SAMIN to allow the agent's developer to improve it. SAMIN uses multi-threading mechanism to assure responsiveness of the SAMIN's GUI during negotiation sessions. Agents running into a deadlock can be stopped by the user by means of the GUI without fatal consequences for the negotiation environment.

The algorithms used in the negotiation strategies can have high computational complexity [19] and, thus, require significant computational power from the execution platform and essential time slot to perform necessary computations to process opponent's offer or select the next action. Negotiation typically, take place under time constraints [5]. Therefore, a timeout mechanism is implemented in SAMIN.

The agents are notified by the negotiation environment about the time left until the deadline using the real-time clock. The timeout mechanism can be switched off by the user when SAMIN is used as a research tool.

3.6 Results

The main advantage of the proposed MAS architecture is to allow for integration of heterogeneous agents and to facilitate comparison of their negotiation. SAMIN can be used as a testbed to perform experiments with various negotiation domains, preference profiles and negotiating agents. Thus, it contributes to automated negotiating agents research by providing a tool that is able to show new insights about such agents. Here we shortly present the most interesting results received with SAMIN for negotiating agents that have been implemented and/or integrated in it.

3.6.1 Experimental Setup

A tournament is a typical experimental setup for evaluation of negotiating agents. It enables analysis of the behaviour and effectiveness of an agent compared to that of others. Multiple negotiation domains and preference profiles can be selected for a tournament. To test sensitivity of a strategy to its internal parameter the value of the parameter can be varied in a tournament. Every session can be repeated a number of times to build a representative sample of negotiation results for a statistical analysis in case of non-deterministic negotiation strategies.

A number of negotiation factors influencing negotiation behaviour have been reported in [11]. We reuse these factors in our method.

Size of the negotiation domain. Complexity of the negotiation domain and preference profiles is determined by the size of the negotiation domain. Size of the domain can influence learning performance of the negotiation strategy and, thus, the outcome reached by the strategy [14]. The size of the domain is exponential with respect to the number of issues. Therefore, to be able to test scalability of a negotiation strategy the experimental setup should have a set of domains ranging from low number of issues to higher number of issues.

Predictability of the preferences. Negotiation strategies can try to exploit the internal structure of the preferences in order to improve one's own efficiency. I.e., the Trade-off strategy assumes that distance measures can be defined using domain knowledge for the preferences of the opponent. These measures combined with the opponent's offers allow the Trade-off strategy to predict opponent preferences and as a result improve efficiency of the bidding. In [11], however, it has been shown that in case of a mismatch of the domain knowledge and the actual structure of the opponent's preferences the performance of a strategy can drastically drop. Therefore, we introduce the notion of the predictability of the preferences into our method.

Issues are called predictable when even though the actual evaluation function for the issue is unknown, it is possible to guess some of its global properties. For example, a price issue typically is rather predictable, where more is better for the seller, and less is better for the buyer, and the normal ordering of the real numbers is maintained; an issue concerning colour, however, is typically less predictable.

Opposition of the preferences. The results of analyzing negotiation dynamics presented in [11] revealed that some negotiation strategies are sensitive to preference profiles with compatible issues. Issues are compatible if the issue preferences of both negotiating parties are such that they both prefer the same alternatives for the given issue. Negotiation strategies may more or less depend on whether preferences of the negotiating parties are opposed or not on every issue. That is, using some strategies it is harder or even impossible to exploit such common ground and agree on the most preferred option by both parties for compatible issues (humans are reported to have difficulty with this as well; cf. [32]). A selection of preference profiles should therefore take into account that both preference profiles with and without compatible issues are included.

To measure the opposition between two preference profile we use ranking distance measure proposed [16]. The measure is based on the conflict indicator proposed in [9]. The conflict indicator function yields 1 when the ranking relation of two arbitrary outcomes based on the utility space of one agent is not the same as the ranking relation based on the utility space of the opponent; if the rankings based on both utility functions match the conflict indicator takes the value of 0. The conflict indicator is calculated for all permutations in the negotiation domain and normalized over the domain. The higher the value of the ranking distance the stronger opposition between the preference profiles.

Another measure for the opposition of preferences proposed in [15] uses Pearson's correlation coefficient for that purpose. This coefficient represents the degree of linear rela-

Domain	Utility spaces		Weights		Domain size	Number of predictable
	Ranking	Pearson	Ranking	Pearson		
AMPO vs. City	0.662	-0.482	0.422	-0.139	7,128,000	3 (10)
Party	0.540	-0.126	0.467	-0.276	3,125	0 (5)
SON	0.669	-0.453	0.833	-0.751	810,000	4 (4)
2nd hand car	0.635	-0.387	0.600	-0.147	18,750	1 (5)
Employment contract	0.698	-0.584	0.600	-0.241	3,125	5 (5)

Table 3.1: Summary of the negotiation domains and preference profiles

tionship between two variables. The Pearson’s correlation coefficient takes a real value from the interval $[-1; 1]$. A value of $+1$ means that there is a perfect positive linear relationship between variables, whereas a value of -1 means that there is a perfect negative linear relationship between variables. A value of 0 means that there is no linear relationship between the two variables.

The following negotiation domains and preference profiles are available in SAMIN (see Table 5.1 for summary):

- The *Second hand car selling* domain, taken from [18], includes 5 issues. Only the buyer’s preferences and the price issue are predictable, in the sense that an agent can reliably predict the other agent’s preferences associated with an issue.
- The *Party* domain is created for negotiation experiments with humans. It is a rather small domain with 5 discrete issues with 5 possible values each. All of the issues are unpredictable. In this domain, the preference profiles used are not as opposed to each other as in the other domains.
- The *Employment contract negotiation* domain, taken from [26] with 5 discrete issues. All issues have predictable values. The preference profiles are strongly opposed, i.e. both negotiators dislike outcomes that the other prefers most.
- The *Service-Oriented Negotiation* domain, taken from [6], includes 4 issues. All issues are predictable, i.e. based on available “domain knowledge” preferences can be reliably predicated.
- The *AMPO vs City* domain, taken from [30], includes 10 issues, of which 8 are predictable. Information about the opponent’s issue priorities, i.e. the weights agents associate with issues. This is a large domain with more than 7,000,000 possible outcomes.

3.6.2 Experimental Results

Here we present the most interesting results we received for the state-of-the-art agents described in Section 3.3.2.

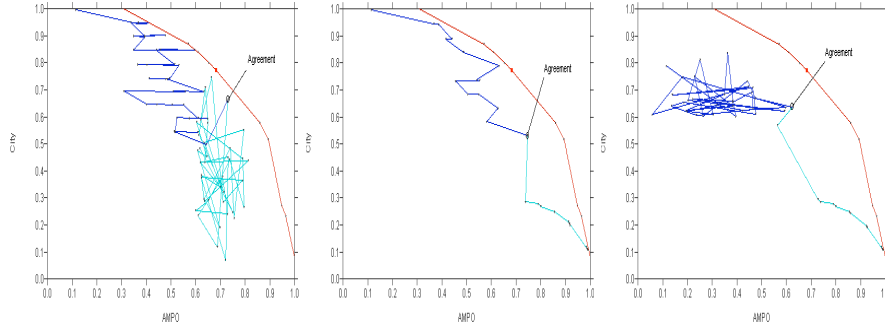


Figure 3.6: Dynamics of negotiation process for: a) Trade-off (City) vs Random Walker strategy (AMPO), b) Trade-off (City) vs ABMP strategy (AMPO), c) Random Walker (City) vs ABMP strategy (AMPO).

Trade-off and ABMP Agents Figure 3.6 shows typical runs in the AMPO vs City domain. Figure 3.6a shows a run of Trade-Off, representing the City, versus Random Walker (with break-off set to 0.6), playing AMPO. The Random Walker strategy is insensitive with respect to its own preferences. This fact, combined with the lack of information of relative importance of issues (weights) causes the unfortunate moves (an offer that decreases utility for both parties compared to the agent's previous offer, see [11]) produced by the Trade-off strategy.

Figure 3.6b shows Trade-off (as City) vs ABMP (as AMPO) in which ABMP is rather insensitive to the behaviour of the opponent, and Trade-off is sensitive. In this domain Trade-off really exploits the available domain knowledge. Figure 3.6c shows Random Walker (City) vs ABMP (AMPO). ABMP always concedes on all issues, determining the size of the concession on the difference between the utilities of its own bid and that of its opponent. It does not use previous opponent bids to get insight into the opponent's preferences and, as a result, does not adapt much to the strategy of the opponent.

This analysis shows a direct link between the correctness and/or completeness of the domain knowledge and opponent preferences sensitivity. The Trade-off strategy is very sensitive to opponent preferences given complete information. In that case, the similarity functions exactly match the opponent's preferences and the weights exactly represent the issue importance factors of the opponent.

The SON domain does not have information about weights of the similarity functions and thus opponent preferences sensitivity of the Trade-off strategy decreases but it is still more sensitive to the opponent preferences than ABMP. Similarity functions for the Second hand car domain were defined in such a way that they often do not match the preferences of the negotiation opponents. In addition, the weights of the similarity function do not match the opponent's importance factors of the negotiation issues. This leads to under performance of the Trade-off strategy while ABMP shows more robust negotiation behavior. The experiments show that if less domain knowledge is available, Trade-off makes more unfortunate steps.

In general, when issues are predictable, the chance of making an unfortunate step becomes small. This aspect becomes clear in the car domain, where the seller's preferences are rather predictable, but the buyer's preferences vary a lot.

We conclude that it is impossible to avoid unfortunate steps without sufficient domain knowledge or opponent knowledge. Indeed, the similarity criteria functions used in the Trade-off Strategy provide general information about the negotiation problem, but do not take into account the specific attributes of the negotiating parties. In any particular case, a negotiator may deviate from the generalized domain model in various ways. Approaches as reported in [4, 23, 32] apply techniques to learn more about the opponent.

Bayesian Agent In small domains such as the SON domain, the Bayesian agent is very efficient in learning issue weights and evaluation functions of the issues that is indicated by the fact that the negotiation trace almost coincides with the Pareto frontier, see [14] for the details. Here we demonstrate the effectiveness of the scalable version of the Bayesian Agent on larger domains. The results on the AMPO vs City domain presented in Figure 3.7 show, as is only to be expected, that it becomes harder to stay close to the Pareto efficient frontier. The performance of the Bayesian learning agents is now similar to that of the agent based on the Trade-off strategy and both stay close to the Pareto frontier. The ABMP strategy shows similar behaviour as on the other negotiation domains, and is outperformed by the other strategies. The results thus are still very good. Also, note that the agreement reached by the Bayesian agents has a higher utility than that reached by the other strategies and that both the Bayesian agent without domain knowledge as well as the Trade-off agent make quite big unfortunate steps.

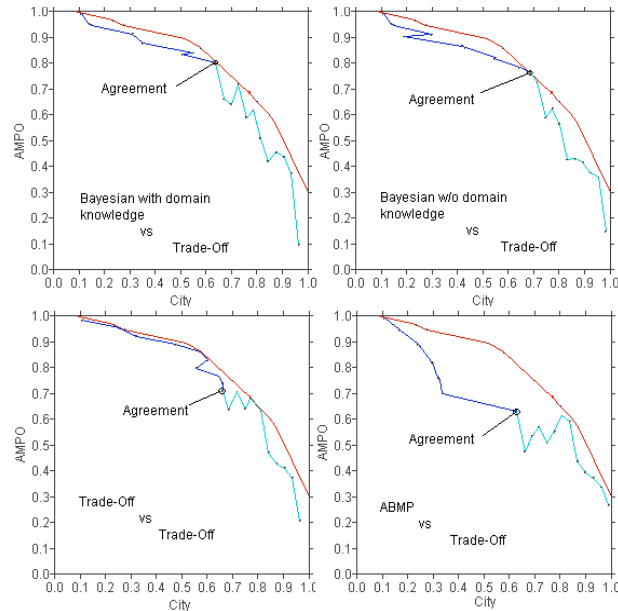


Figure 3.7: Negotiation dynamics for the Bayesian agent on the AMPO vs. City domain

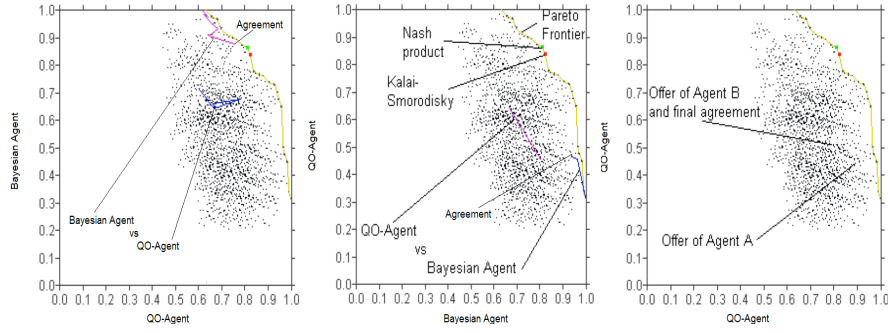


Figure 3.8: Negotiation dynamics for the QO agent on the Party domain

QO Agent Figure 3.8 presents the results of the negotiation experiment. A small and simple negotiation problem, called “Party” [14], is used to analyze the performance of the QO agent within our negotiation framework. This domain has been created for negotiation experiments with humans, which also explains its rather limited size. The charts show the space of all possible negotiation outcomes. The axis represent the utilities of the outcomes with respect to the utility functions of the negotiating agents. The charts show the negotiation paths of the agents marked by arrows with the names of the agents.

The Bayesian agent starts with an offer that has maximum utility. It tries to learn the opponent preferences from the offers it receives and uses this model when it makes a concession towards the opponent. As a result, it stays close to the Pareto Efficient frontier. The QO agent in this domain has more difficulty to propose efficient offers. This is a result of limitation of the opponent model of the agent. The QO agent accepts an offer of the Bayesian agent as soon as such an offer has a utility level for the QO agent that is higher than utility of the QO agent’s own offer.

Fuzzy-based Model Agent The other agent integrated into SAMIN is the FBM agent introduced in [29]. The FBM agent was tested in a setup where it has to negotiate against the Bayesian agent about a single issue defined on real values ranging from 10 to 30. The original FBM agent is designed for negotiations where agents can exchange fuzzy proposals. The implementation of the FBM agent we used is able to negotiate about one-issue negotiations but can be extended for multi-issue negotiations. The agent adopts time dependent negotiation tactics from [5] and, thus, always makes concessions towards opponent. The offers are defined using two values: the peak value and the stretch of the offer. The preference profiles of the agents used were in complete opposition: the FBM agent wants to minimize the value of the issues and the Bayesian agent tries to maximize it. In the experiments we performed, the β parameter that defines whether an agent makes bigger concessions in the beginning of the negotiation (Conceder) or at the end (Boulware) was varied, see Table 3.2.

In a single issue negotiation there is no possibility for a “win-win” outcome and all negotiation outcomes are Pareto efficient. One of the more important aspects of a negotiation strategy for a single issue negotiation is how fast it concedes to the opponent. As a re-

Agents	Utility						
	$\beta=0.02$	$\beta=0.1$	$\beta=0.5$	$\beta=1$	$\beta=2$	$\beta=10$	$\beta=50$
FBM Agent	0.898	0.897	0.734	0.585	0.449	0.193	0.060
Bayesian Agent	0.102	0.103	0.266	0.415	0.551	0.807	0.940

Table 3.2: Utility values of the FBM and Bayesian agents

Negotiation Domain	Negotiation Strategy			
	ABMP	Trade-Off	Bayesian Smart	Bayesian Tit-for-Tat
Car	16%	12%	13%	14%
Party domain	13%	9%	13%	14%
Service-Oriented	14%	17%	25%	38%
Employment contr.	11%	40%	44%	47%
AMPO vs City	10%	13%	14%	20%

Table 3.3: Increase in utility for the Bayesian Tit-for-Tat strategy relative to the Random Walker strategy

sult, for $\beta > 1$ the FBM agent implements a Conceder tactic and the FBM agent under performs with respect to the Bayesian agent that makes linear concessions in this case because no moves towards the Pareto frontier are possible. When the FBM agent employs a Boulware tactic ($\beta < 1$) the Bayesian agent starts conceding significantly and the result is a much lower utility for the Bayesian agent.

Bayesian Tit-for-Tat Agent As discussed, the main objective associated with a negotiation strategy is to gain the best agreement possible in a negotiation. Utility of an agreement, therefore, measures the efficiency of a strategy. For every negotiation domain and preference profile the utility of agreements achieved by a strategy were averaged over all opponent strategies in the tournament. We assume that an efficient negotiation strategy should perform better than the Random Walker strategy. Therefore, we calculate the percentage of the utility increase compared to the utility of the Random Walker strategy (see Table 3.3).

The results show that on all domains the Bayesian Tit-for-Tat strategy performs better than all other strategies currently available in the negotiation repository. Only on the 2nd hand car negotiation domain the Bayesian Tit-for-Tat strategy is outperformed by the ABMP strategy. As in this domain a concession-based strategy is very efficient, and ABMP aims to concede on all issues, this strategy does particularly well in this domain.

The most significant increase in the efficiency of the reached agreement is shown on the Employment contract negotiation domain. This negotiation domain is rather small and evaluations of the issue alternatives are predictable in this domain. Learning in such a domain is relatively simple and, as a result, the Bayesian Tit-for-Tat strategy shows excellent performance. The Trade-off strategy shows good performance as well, however, it does not perform as well as the Bayesian Tit-for-Tat strategy. The ABMP strategy is

significantly less efficient than the Bayesian Tit-for-Tat and the Trade-off strategies due to presence of issues with compatible preferences.

Similar results are obtained for the Service-Oriented Negotiation domain. This domain is much bigger than the Employment contract domain in terms of the possible agreements but has less issues. In addition, weights of the issues in the SON domain have bigger variation then in the Employment Contract domain where importance of the issues is more uniform. This explains the much lower efficiency of the Trade-Off strategy that is not capable of dealing with the weights of the issues. The Bayesian Tit-for-Tat strategy learns weights of the issues in the opponent preference profile and therefore shows a better performance.

AMPO vs City domain is the biggest domain in the repository. As is to be expected, the performance of the learning technique used in the Bayesian Tit-for-Tat strategy degrades in such bigger negotiation domains. This explains the lower relative increase in Table 3.3.

3.6.3 Approximating Auction Mechanism with Negotiation

In Section 3.4.2 we introduced a one-to-many negotiation protocol that approximates an auction mechanism. Here we present experimental results received for the proposed negotiation protocol. Figure 3.9 shows the histograms of the differences in utilities between the outcomes received with the original auction mechanism and the negotiation protocol.

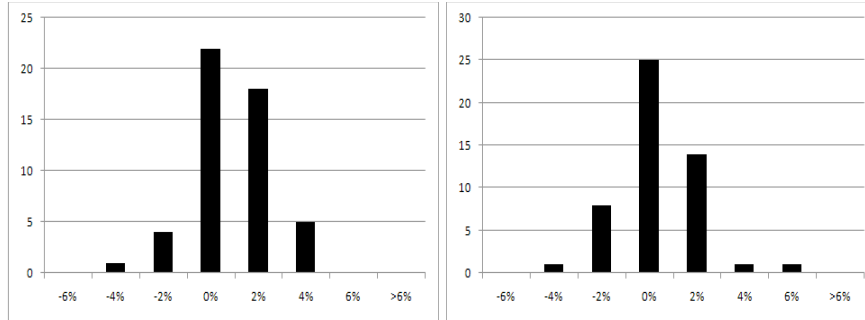


Figure 3.9: Histograms of the differences in the utilities of experimental and theoretical outcomes for the buyer (left) and the seller (right).

The winner predicted by the mechanism and the negotiation protocol coincide 100%. This means that the negotiation protocol does not change the results of the first round in which a seller is selected as winner. Moreover, in the second round, in general the outcomes obtained by negotiation are also quite close to those determined by the mechanism. That is, in 78% of the experiments the deviation is less than 5%. The standard deviation of the difference between the mechanism outcome and the experimental results is 4%, and in 94% of the experiment the deviation did not differ with more than 10%, indicating that overall outcomes were reasonably close to the mechanism outcome with a few exceptions.

This means that the negotiating agents that can learn are able to approximate the outcome determined by the mechanism quite well.

3.7 Conclusion

SAMIN, the system for analysis of multi-issue negotiation introduced here, has proved to be a valuable tool to analyse the dynamics of human-human closed negotiation against a number of dynamic properties. Our analysis shows that humans find it difficult to guess where the Pareto Efficient Frontier is located, making it difficult for them to accept a proposal. Although humans apparently do not negotiate in a strictly Pareto-monotonous way, when considering larger intervals, a weak monotony can be discovered. Such analysis results can be useful in two different ways: to train human negotiators, or to improve the strategies of software agents. Clear from our research so far, is that five key factors shape the outcome of a bilateral negotiation with incomplete information: (i) knowledge about the negotiation domain (e.g. the market value of a product or service), (ii) one's own and one's opponent's preferences, (iii) process attributes (e.g. deadlines), (iv) the negotiation strategies, and (v) the negotiation protocol.

The use of agent technology for negotiation systems has been a big help in both the design and the implementation of the SAMIN system. Principled design methods for agents and multi-agent systems such as DESIRE ensured a transparent design that properly reflects the interests of the stakeholders (researchers) and negotiators (human and software agent). The organization makes it easy to run tournaments with any number of agents, and over a number of negotiation domains. The interface and adapters to connect agents to the negotiation environment have been clearly specified which enable an easy integration of heterogeneous negotiating agents. The graphical user interfaces support both researchers and human subjects participating in experiments.

A good start has been made in the development of a toolkit for analysis in SAMIN, but more work needs to be done. Additional research on ontologies for negotiation is required to make this feasible; for example, we cannot currently formulate associated constraints on the domain of negotiation that must be satisfied for an agreement to be acceptable. More technically, components for web integration as well as extensions of adapters need to be developed, e.g., in order to handle more generic ontologies.

Bibliography

- [1] R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In *The 3rd Int. Central And Eastern European Conf. on Multi-Agent Systems*, 2003.
- [2] T. Bosse, C.M. Jonker, and J. Treur. Experiments in human multi-issue negotiation: Analysis and support. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the Third International Joint Conference on Autonomous*

Agents and Multi-Agent Systems, AAMAS'04, page 672 – 679. IEEE Computer Society Press, 2004.

- [3] F.M.T. Brazier, B.M. Dunin-Keplicz, N.R. Jennings, and J. Treur. Formal specification of multi-agent systems: a real world case. *International Journal of Co-operative Information Systems, IJCIS*, 6(1):67–94, 1997.
- [4] R.M. Coehoorn and N.R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings of the 6th International Conference on E-Commerce*, pages 59–68, 2004.
- [5] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [6] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
- [7] R. Fisher and W.L. Ury (and for the latest edition B. Patton). *Getting to Yes: Negotiating Agreement Without Giving In*. Penguin Books, 1981, 1992, 2003.
- [8] Dhananjay K. Gode and Shyam Sunder. Allocative efficiency in markets with zero intelligence (zi) traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.
- [9] Vu Ha and Peter Haddawy. Similarity of personal preferences: Theoretical foundations and empirical analysis. *Artificial Intelligence*, 146(2):149–173, 2003.
- [10] Paul Harrenstein, Tamas Mahr, and Mathijs M. de Weerd. A qualitative vickrey auction. In Ulle Endriss and Goldberg Paul W, editors, *Proceedings of the 2nd International Workshop on Computational Social Choice*, pages 289–301. University of Liverpool, 2008.
- [11] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Negotiation dynamics: Analysis, concession tactics, and outcomes. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*, pages 427–433, 2007.
- [12] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Using opponent models for efficient negotiation (extended abstract). In Sierra Decker, Sichman and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, 2009.
- [13] Koen Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov. Towards an open negotiation architecture for heterogeneous agents. In *Proceedings of 12th International Workshop CIA 2007 on Cooperative Information Agents*, Lecture Notes in AI. Springer-Verlag, 2008.
- [14] Koen Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the AAMAS 2008*, 2008.

- [15] Koen Hindriks and Dmytro Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. In *Proceedings of the AMEC 2008*, 2008.
- [16] Koen V. Hindriks, Dmytro Tykhonov, and Mathijs de Weerd. Approximating an auction mechanism by multi-issue negotiation. In Koen V. Hindriks and Willem-Paul Brinkman, editors, *Proceedings of the First International Working Conference on Human Factors and Computational Models in Negotiation (HuCom2008)*, pages 33–38, December 2008.
- [17] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
- [18] C.M. Jonker and J. Treur. An agent architecture for multi-attribute negotiation. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on AI, IJCAI'01*, pages 1195 – 1201, 2001.
- [19] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Paper 125 of the Center for eBusiness@MIT*. <http://ebusiness.mit.edu>, 2001.
- [20] R. Kowalczyk and V. Bui. On constraint-based reasoning in e-negotiation agents. In F. Dignum and U. Cortés, editors, *Agent-Mediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce Systems*, Lecture Notes in Computer Science, pages 31–46. Springer Verlag, 2003.
- [21] Guoming Lai and Katia Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18(2):169–187, March 2009.
- [22] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall PTR, 3 edition, 2004.
- [23] R. Lin, S. Kraus, J. Wilkenfeld, and J. Barry. An automated agent for bilateral negotiation with bounded rational agents with incomplete information. In *Proc. of the 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 270–274, 2006.
- [24] Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence Journal*, 172(6-7):823–851, 2008.
- [25] R.H. Mnookin, S.R. Peppet, and Andrew S. Tulumello. *Beyond Winning: Negotiating to Create Value in Deals and Disputes*. Harvard University Press, 2000.
- [26] J. Nadler, L. Thompson, and L. van Boven. Learning negotiation skills: Four models of knowledge creation and transfer. *Journal of Management Science*, 49(4):529–540, 2003.
- [27] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

- [28] D.G. Pruitt. *Negotiation Behavior*. Academic Press, 1981.
- [29] Zaynab Raeesy, Jakub Brzostowski, and Ryszard Kowalczyk. Towards a fuzzy-based model for human-like multi-agent negotiation. In *Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, pages 515–519, 2007.
- [30] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2003.
- [31] T. Sandholm. *Multi-agent Systems: A Modern Introduction to Distributed Artificial Intelligence*, chapter Distributed rational decision making. MIT Press, 1999.
- [32] L.L. Thompson. *The Heart and Mind of the Negotiator*. Pearson Prentice Hall, 2005.
- [33] D. Zeng and K. Sycara. Benefits of learning in negotiation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997.
- [34] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human Computer Systems*, 48:125–141, 1998.

Chapter 4

BOP: an Effective Bayesian Learning Algorithm for Opponent Preferences in Negotiation

An important class of real-life negotiations is the class in which people negotiate once with one opponent on a particular topic. The more one knows about the opponent, the better the negotiation result will be. To improve bidding support systems in this paper we present a generic learning algorithm that is based on Bayesian learning techniques, that is capable of learning the most important aspects of the opponent's profile during the bidding in one negotiation session, i.e., ranking of the weights of the issues and approximations of the evaluation functions per issue. For large domains we present a slightly less effective, but scalable variant of the algorithm. Available domain knowledge, and / or information about the opponent can be integrated to give the learning process a head start. Our algorithm proved itself in a rigorous and extensive test. Opponent profiles can vary from cooperative to competitive, and their preferences can vary from simple to complex. It is robust with respect to a range of benchmark strategies played by autonomous agents and also when the opponent is human.

4.1 Introduction

Negotiation is a type of interaction used by self-interested parties to reach a mutual agreement about a common decision to be made. To reach an agreement negotiation parties exchange various offers or bids using e.g. an alternating offers protocol [18]. In reaching such an agreement both parties 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. Such agreements can be complex and contain multiple issues to be settled. If multiple issues are discussed, differences in the parties' preferences allow for agreements

mutually beneficial for both parties, so-called *win-win* outcomes [19]. Such a negotiation is therefore not an adversarial zero-sum game.

In every negotiation learning about the opponent and his preferences is fundamental for reaching a good outcome. Effective bidding strategies are nice illustrations of this argument, as the strategy used to make negotiation moves is an important factor in the quality of outcome. A number of negotiation strategies have been proposed in the literature, e.g., see [4, 17, 9, 13]. Given the point that the opponent has to be satisfied as well in order for a negotiation to end in a deal, all strategies proposed in the negotiation literature at some moments try to make concessions. A concession is a bid that for the one making the bid has a decreased utility compared to the previous offer. However, for the opponent to feel it as a concession, the bid should also have an increased utility for the opponent. This is complicated, by the fact that negotiating parties are generally not willing to reveal their preferences in order to avoid exploitation. However, a good strategy has to defend the preferences of the party using the strategy. If the negotiating parties end up with a deal that could be improved for some without hurting the others, then the negotiators leave money on the table, and the deal is called Pareto Inefficient. The set of bids that cannot be improved for at least one of the negotiation parties, without making it worse for the others is called the Pareto Efficient Frontier. Without good information about the preferences of the other party it is impossible to decide on a good negotiation move and to reach a Pareto Efficient agreement. Given the fact that in general parties are not willing to reveal their preferences, the good negotiator uses techniques for learning about the opponent and his preferences.

To deal with the problem of incomplete information a number of learning techniques have been proposed for the negotiation domain. A common assumption in the literature on learning in negotiation is that some historical data is available which can be used to model an opponent's behavior. For example, in [12] authors estimate the probability of acceptance of an offer by an opponent using information about accepted and rejected offers by the opponent in the previous negotiations. If such information is available, this is an excellent approach. In real-life negotiations, however, the people involved might never have negotiated with each other before, and if they did, then not necessarily on that same topic. In such cases the opponent's profile cannot be learned from historical data and other learning approaches will have to be developed.

There is a little research about how to learn opponent preferences in a closed single-session negotiation, e.g. see [3]. A complicating factor in this context is that the number of moves performed before reaching an agreement is limited (typically about 5 to 30 moves when the negotiators are human), and individual bids do not provide much information [21]. To deal with this problem in the literature some restrictive assumptions are made, such as a strong opposition of the preferences, or efficiency of an opponent's strategy. Such assumptions are shown to be valid in certain domains, but not in all. Therefore, a learning technique designed and tested in a specific negotiation domain may not be applicable to another negotiation domain [8]. The purpose of this work is to design a generic leaning technique to model the opponent's preferences in a closed single-session multi-

issue negotiation. Based on the analysis of [8] we formulate a number of requirements for a generic learning technique to be used in real-life negotiations.

The first requirement is that the kernel of the learning algorithm should be domain and opponent independent. The number of issues and values per issue in real-life negotiation domains can change during the negotiation and from one negotiation to another. For example, the issues for negotiating about real estate differ per set of negotiators, and new regulations (e.g., energy conservation properties of the object) have their influence. Even if the issues and possible values per issue would be stable, still the preferences can have significant variation over the negotiation opponents. Therefore the learning algorithm should not depend on any features of a negotiation domain or negotiation opponent in its design that can decrease its learning performance in other domains. Given a domain specification the learning algorithm must be able to start learning with the first offer received. Thus, the essence of the learning algorithm should be domain and opponent independent.

The second requirement is that the learning algorithm should be able to take advantage of domain or opponent specific information. In some cases reliable domain knowledge about typical opponent preference profiles is available or can be learned from historical data. Such knowledge can be used to speed up the learning progress or increase accuracy of the learned preference profile. Therefore, we require the learning algorithm to be able to take advantage of the available knowledge about the domain and opponent.

The third requirement is that the learning algorithm should be robust with respect to incorrect available knowledge about either the domain or opponent. Knowledge that is typically correct for a certain negotiation domain can be wrong for a specific case or sub-domain. For example, the negotiation domain of classic cars differs essentially from the typical domain of negotiating about second hand cars. Similar arguments hold for negotiators that love their classic cars versus a negotiator that is just selling a classic obtained through an inheritance. Ideally the learning algorithm recognizes such a situation. In any case the algorithm should still learn the preference profile of the opponent even if it was initialized with incorrect information.

In this paper we propose an effective Bayesian learning algorithm for Opponent Preferences (BOP) in closed single-session multi-issue negotiation that is designed with respect to the requirements outlined above. The learning technique can be integrated in any negotiation strategy to increase the efficiency of its offers. In other words, given a utility of the next offer BOP can be used to generate a corresponding approximate Pareto efficient offer. For example, the QO strategy of [13] decides on a negotiation move using three predefined opponent's profiles. In a case when the opponent's preferences profile does not belong to these profiles the offers generated by QO strategy would not be efficient. Instead, the QO strategy can use the proposed learning technique to model the opponent's preference and initialize it with the three possible profiles.

In BOP 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 framework is based on Bayesian learning techniques. The algorithm exploits certain

structural features and rationality principles of the bidding process. The framework allows for the incorporation of prior available domain and opponent knowledge but does not require any such knowledge. It thus extends and generalizes previous work on learning in negotiation by introducing a technique to learn opponent preferences for multi-issue negotiation.

The size of the hypotheses space in the standard BOP algorithm is exponential in the number of issues. As such, the standard algorithm is not scalable with respect to the number of issues. Therefore, we developed and tested a simplified version of the BOP algorithm, called ScalableBOP in which the number of hypotheses is linear in the number of issues. The difference is that the probability per hypothesis for one issue is updated independently of the probabilities for the other hypotheses. The speed of the learning and the final quality of the learned profile is less for ScalableBOP, but the effect is still highly significant.

To gain a good understanding of the performance of the proposed learning techniques and the potential to improve the performance of other learning techniques in the context of automated negotiation a systematic assessment method for the quality of learning is presented. It provides the technical tools for analysis, identifies the key factors that need to be taken into account and proposes an experimental setup to evaluate the quality of learning. The technical tools used as well as the approach for analysis are discussed, and we apply the method to BOP to show the effectiveness of BOP. This application of the evaluation method shows some of the insights that may be gained by using it.

This paper is organized as follows. Section 4.2 discusses related work in the area of learning in negotiation. In Section 4.3 we present and motivate our design choices. Section 4.4 introduces the learning technique. In Section 4.5 we introduce measures for quality of opponent preferences learning. Section 4.6 explains characteristics of negotiation domains and preference profiles. Section 7.5 presents experimental results to demonstrate the effectiveness of the approach in various negotiation setups. Finally, Section 7.6 concludes the paper and suggests several directions for future research.

4.2 Related Work

The subject of learning in negotiation received significant attention in the literature. In this section we discuss the possibility of applying various generic learning approaches to our problem and present the state-of-the-art work in the area of learning in the negotiation domain.

4.2.1 Learning Approaches

Over the years a variety of learning techniques has been proposed. Applicability of a specific learning technique is constrained by its characteristics such as a representation of

training set, size of the training set, size of the hypothesis space. In this section we discuss the learning techniques with respect to their applicability and of learning preference profiles from bids exchanged during negotiation only.

Artificial neural networks (ANNs) are based on the biological learning systems consisting of complex webs of interconnected neurons. Feedforward networks containing three layers of units are able to approximate any function to arbitrary accuracy, given a sufficient number of units. Therefore, they are considered to be *universal approximators* [16]. However, learning algorithms required to find appropriate parameters settings of a network typically require a substantial amount of training data. Another problem of the ANNs is overfitting of the input data, i.e., learning of a random noise instead of underlying relationship. ANNs have been successfully applied to the problem of learning an opponent's profile in a setup where a significant number of stored negotiations with a given opponent is available, see [2]. However, ANNs cannot be used to approximate the preferences of the opponent in a single negotiation due to the small number of offers typically exchanged in the negotiation and the lack of information about actual utilities associated with that offers.

Another family of powerful learning algorithms is that of the genetic algorithms (GA). The idea for GA comes from the evolution of biological organisms [16]. GA is typically used for optimization problems where the optimized function is complex containing multiple optima. In GAs a population of abstract representations of candidate solutions to an optimization problem evolves toward better solutions. Similar to ANNs the GAs require substantial amount of training data. For GAs it must be possible to calculate a fitness function for any solution that represents quality. GAs have been successfully applied in experimental setups to increase the efficiency of negotiation strategies in a specific circumstances. For example, an approach proposed in [15] aims at combining different negotiation tactics from [4] in a single strategy because no single tactic or decision function seems to be "right" in an arbitrary negotiation settings. A GA is used to compute a next offer that adjusts the weights associated with each of the individual tactics. The weights are used to combine the relative contribution of each tactic in determining the next offer. Instead of learning an opponent model this approach requires a substantial number of negotiations to learn appropriate weights associated with the tactics. This learning approach moreover requires that the preference profiles of both parties are made public in order to calculate the fitness during the learning phase. As a result, the weights learned to combine the strategies only yield efficient negotiations in specific negotiation setups. This makes GA less suitable for our problem.

An expressive and human readable representation for learned hypotheses is in the form of sets of *if-then* rules. Such learning is often called inductive logic programming (ILP), cf., [16]. ILP is based on a subset of first-order logic, more precisely rules containing variables. A variety of algorithms have been proposed to learn such rules from training examples. Learning of the opponent preferences takes place under uncertainty about the opponent's strategy. To deal with uncertainty ILP requires additional extensions of the hypotheses space and learning algorithms. Furthermore, the choice of the hypotheses

representation can have negative consequences for the computational tractability of the learning. Hence, mapping of the preference profile learning problem on the hypotheses space used by ILP is not trivial and applicability of the ILP approach for learning in negotiation requires additional investigation.

Bayesian learning algorithms are among the most practical approaches to certain types of learning problems. Unlike the algorithms (e.g., ILP) that completely eliminate a hypothesis if it is found inconsistent with any single example in Bayesian learning each observed training example can incrementally increase or decrease the estimated probability that a hypothesis is correct. This provides a flexible learning approach that is useful in cases when reasoning about the hypothesis is performed under a great degree of uncertainty. For example, in literature to learn the opponent's preferences some assumptions are made about the opponent's tactics. While such an assumption might be correct in general it would not be possible to reproduce the opponent's tactics exactly.

To evaluate a new instance (in our problem that would be an opponent's offer) the Bayesian learning combines predictions of multiple hypotheses, weighted by their probabilities. As a result, it is not required that at least one hypothesis in the hypotheses space accurately reconstructs the actual preference profile of the opponent. Instead, the opponent's profile that is not explicitly represented by a single hypothesis in the proposed hypotheses space can be approximated by a combination of several hypotheses (see Section 4.3.1 for the details). Therefore, the range of preference profiles that can be learned is much richer than the profiles represented by the hypotheses space.

Even in cases where Bayesian methods prove computationally intractable, they can be simplified by assuming independence of the factors to be learned. Prior knowledge can be combined with observed data to increase accuracy of the learned model and increase the speed of learning. These considerations make Bayesian method suitable for our problem.

Among the learning techniques considered in this section Bayesian learning seems to be the most suitable to our needs. ANNs and GAs typically require substantial amount of training data. In a single-session negotiation, however, there is not enough interaction that can be used as training data. Unlike other learning techniques Bayesian learning even functions if only a few offers are available to learn probabilities of the possible opponent's preference profile. Unlike conventional ILP Bayesian learning can handle noisy and inconsistent training data. The representation of the hypotheses is flexible, allowing a wide range of opponent's preference profiles that can be learned.

4.2.2 Learning in Negotiation

One interesting approach to learning efficient negotiation strategies is based on building a training data set from information about multiple sessions, see [2], [12]. For example, Carbonneau and co-authors proposed ANN to learn a strategy that would generate efficient negotiation offers [2]. The ANN is trained using a database of stored negotiation sessions between humans. Narayan et al. (2008) presented an approach to learn an oppo-

nent's negotiation strategy as a sequence of bids made by that party. The approach uses Markov chains to model the opponent strategy and Bayesian learning to update the probabilities of the transitions between states in the Markov chain. It does assume, however, that negotiations involve only *one issue*. Automated learning of a negotiation strategy is hard and is only feasible using data from *multiple, past negotiations*. The more information about past negotiations on the same domain available the better the learning performance.

The learning approaches described above are typically designed for a specific negotiation domain and require substantial data about negotiation sessions. Furthermore, the approaches lack the possibility of adaptation to the features of a specific opponent or to changes in the negotiation domain. While the approaches can be efficient in situations for which they were trained they cannot be directly applied to other situations without re-design. For example, an ANN would require re-design of the inputs and outputs of the network, learning of the weights of the nodes, etc.

A few negotiation strategies have been built that try to guess opponent's preferences in a single session. A natural suggestion for that is to try and incorporate additional knowledge into a negotiating agent, see e.g., [9]. The effectiveness of providing knowledge about the domain of negotiation has been demonstrated in the Trade-off strategy introduced in [5]. In particular, this paper shows that domain knowledge (coded as so-called *similarity functions*) can be used to select bids that are close to an opponent's bids, thus increasing the likelihood of acceptance of a proposed bid by that opponent. In this approach, the knowledge represented by similarity functions is assumed to be public. As is to be expected, if similarity functions can be found, the Trade-off strategy outperforms a concession-based strategy such as ABMP [10], see [8] for the analysis details. Incorporating public domain knowledge into a strategy, however, still does not take into account the private preferences or priorities that an opponent associates with negotiated issues. The more knowledge of these preferences is available the better the chance of win-win scenarios and optimal outcomes.

Another learning approach in negotiation is based on the assumption that a *fixed set of possible opponent profiles* is given. Bayesian learning techniques can determine the likelihood that an opponent has one of these given profiles. The profile types are assumed to be public knowledge and an agent only has to learn which type of profile its opponent most likely has. The QO agent proposed in [13] is an example of this approach (see also [22, 5] for similar approaches). In the QO agent three opponent profiles are defined and a probability is associated with each of the possible opponent profiles. An update mechanism interprets the observed offers from the opponent and updates the probability distribution according to the opponent strategy model. Then a qualitative decision making that combines information about own preferences and the most probable opponents preferences is used to make an offer. In [14] the QO agent is extended with a learning mechanism that uses a database of past negotiation sessions. The agent performs offline learning based on the kernel based density estimation and using the database as a training set. The results of learning allow the agent to attach acceptance probabilities to each pos-

sible agreement and then use these probabilities in its decision making component, either when proposing a new offer or when determining its concession size.

4.3 Motivation for Design Choices

Our goal is to introduce a learning approach that can be used to model an opponent in a negotiation with imperfect information. To reach this goal a number of design choices for our learning technique have to be made. In this section, we firstly motivate our design choices and secondly, explain them in details.

Negotiation can be viewed as an instance of a Bayesian game. In game theory, the class of Bayesian games refers to games in which players do not have complete information about each others' preferences (or *types*) [18]. In such a setting, players can use evidence (or so-called *signal functions*) to update their beliefs about the other party. In a Bayesian game, in order to be able to learn, it is necessary to specify the strategy spaces and type spaces. Ideally, these spaces are defined generically enough to allow learning of a rich variety of opponent profiles. At the same time, however, these spaces should not be so rich to make it impossible to learn an opponent profile from the limited available evidence (in our case, the opponent's bids).

The complicating factor for learning an opponent's preference in a closed multi-issue negotiation is that the amount of information exchanged during negotiation is minimal. Typically, only offers are exchanged between the negotiating parties with no utility information attached. When a negotiating party sends an offer to the opponent it proposes a potential agreement (if accepted by the opponent) from which we can conclude that the offer is acceptable for the party. Because negotiation parties are eager to reach a deal it would be reasonable to adopt some kind of concession-based negotiation tactic. Typically, in the automated negotiation literature concession-based strategies have been proposed, see e.g. [4, 9]. This line of reasoning is used to make a model of the opponent's negotiation tactic in our proposed learning technique.

The number of the exchanged offers in human negotiations in a single session is small $\tilde{U}$ around 7-8 offers, see e.g. [21]. Domain knowledge about negotiating issues can be used to assess the predictability of an issue.

In this section, we present the hypothesis space that defines the range of opponent profiles that can be learned. We do so by introducing various reasonable design choices about the structure of opponent profiles as well as about an opponent's negotiation strategy. These choices are made to ensure the task of learning an opponent model is feasible. In Section 7.5 we present evidence that the proposed model is both effective as well as rich enough to learn opponent preferences in various negotiation domains.

4.3.1 Structure of the Opponent's Preferences

Our first design choice is related to the representation of the opponent's preferences. It is common to assume that the utility of a bid can be computed as a weighted sum of the utilities associated with the values for each issue, see e.g., [19]. Utility functions modelling the preferences of an agent thus are linearly additive functions and are defined by a set of weights w_i (or priorities) and corresponding evaluation functions $e_i(x_i)$ for each of n issues by:

$$u(b_t) = \sum_{i=1}^n w_i e_i(x_i \in b_t) \quad (4.1)$$

where x_i is the value of issue I in bid b_t in the negotiation round t . To ensure that a utility function has a range in $[0, 1]$, the range of the evaluation functions is assumed to be in $[0, 1]$ and the weights are assumed to be normalized such that their sum equals 1.

In order to learn an opponent's preference profile or utility function $U(b)$ we need to learn both the issue weights w_i as well as the evaluation functions $e_i(x_i)$. The objective of learning an opponent model thus is to find a model as defined by equation (4.1) that is the most plausible candidate or best approximation of the opponent's preference profile.

Our next design choice concerns the issue weights in a preference profile characterised by equation (4.1). Some knowledge about issue weights is important in order to be able to propose a trade-off on issues that are valued differently by negotiating parties. In [8] it is shown that in general it is not sufficient to know issue preferences, i.e., evaluation functions $e_i(x_i)$, to be able to make trade-offs. Making trade-offs is an important means to get closer to the Pareto efficient frontier. To be able to propose a trade-off an agent, say S must know at least two issues one of which, say A , which is valued more by itself than by its opponent and one, say B , which is valued more by the opponent than by S itself. In that case, S can make a concession on B and propose a value for A that is more highly valued by S itself.

In [5] and [9] it is argued that it is sufficient to know the ranking of the weights to be able to make trade-offs and significantly increase the efficiency of an outcome. We propose to define the set of hypotheses H^w 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 $h_j \in H^w$ about weights, which can be computed as a linear function of the rank and also ensures weights are normalized, as follows:

$$w_i = 2 \frac{r_i^j}{n(n+1)} \quad (4.2)$$

where r_i^j is the rank of weight w_i in the hypothesis h_j and n is the number of issues.

Finally, we need to impose some additional structure on the evaluation functions 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 types. A third design choice thus concerns the shape of evaluation functions. We three types of evaluation functions to model preferences over issue values:

1. *downhill* shape: minimal issue values are preferred over other issue values (think, e.g., of price and delivery time for a buying agent), and the evaluation of issue values decreases linearly when the value of the issue increases;
2. *uphill* shape: maximal issue values are preferred over other issue values (think, e.g., of price and delivery time for a selling agent), and the evaluation of issue values increases linearly when the value of the issue increases;
3. *triangular* shape: a specific issue value somewhere in the issue range is valued most and evaluations associated with issues to the left ("smaller") and right ("bigger") of this issue value linearly decrease (think, e.g., of an amount of goods).

Figure 4.1 below illustrates this set of functions and introduces labels $h_{i,j}^e$ to refer to the hypothesis that issue i has associated evaluation function j .

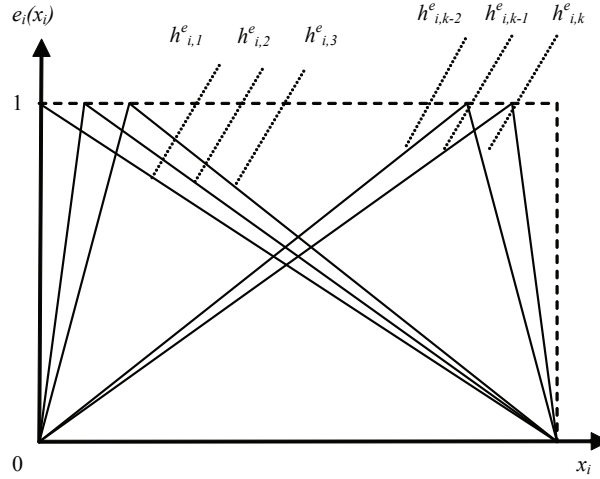


Figure 4.1: Hypothesis space of possible evaluation functions.

The three function types that define the range of possible evaluation functions are common in the literature, and, most importantly, in combination allow for the modelling of other types of function as well (see Figure 4.2 below).

In order to see this, it should be taken into account that a probability distribution is associated with each hypothesis. This allows other types of functions to be approximated by associating different probabilities with various hypotheses. The predicted evaluation of an issue value is derived from all hypotheses that are assigned a non-zero probability. The evaluation thus can be viewed as computing a most probable evaluation value of an

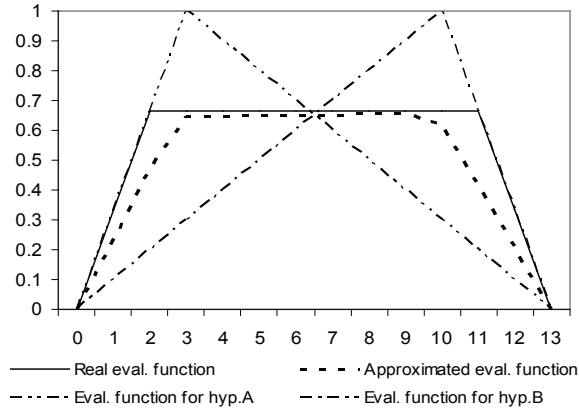


Figure 4.2: Approximation of an evaluation function that is not in the hypothesis space by means of two evaluation functions.

issue value by computing the weighted sum of all evaluations of an issue value associated with some hypothesis with non-zero probability. Different probability distributions thus allow for approximating different types of evaluation functions that do not need to match any single evaluation function from the hypothesis space. Figure 4.2 shows an example of the approximation of a more complex evaluation function (solid line) that is not present in the hypothesis space. Many complex evaluation functions thus can be successfully approximated by a composition of several simple evaluation functions from the hypothesis space. The preferences of an agent can be viewed as a membership function that assigns a degree of membership to each hypothesis in the hypothesis space similar to membership in fuzzy set theory. In our case the membership of an evaluation function is modelled as a probability distribution and our approach is similar to that of triangular membership functions [17].

To summarize, the set of hypotheses concerning an opponent's preference profile is a Cartesian product of the hypotheses about issue weights H^w and shapes of issue evaluation functions $H_i^e : H = H^w \times H_1^e \times H_2^e \times \dots \times H_n^e$.

4.3.2 Model of Opponent's Strategy

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 want to compute $P(h_j|b_t)$ where b_t is the bid proposed at time t . To be able to use Bayes' rule to do this, we need some information about the utility the opponent associates with bid b_t .

As this information is not generally available, we need to make a design choice about a model of the opponent's strategy to be able to make an educated guess of the utility value of b_t for an opponent. In the model we assume that our opponent follows a more or less rational strategy in proposing bids. In particular, we will assume that an opponent follows some kind of concession-based strategy. Although assuming such behaviour may not always be realistic it typically is necessary to perform at least some concession steps in order to reach an agreement. Moreover, in game-theoretic approaches and in negotiation it is commonly assumed that agents use a concession-based strategy [5, 18].

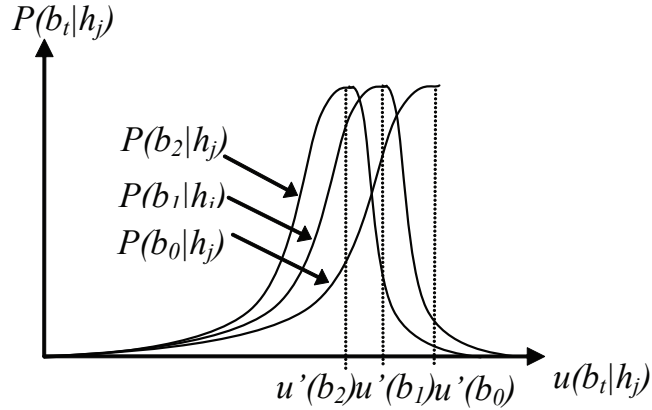


Figure 4.3: Conditional probability distribution of tactics.

In line with [4] we model the opponent's strategy using a time-dependent tactic (TDT). It starts with a bid of a maximal utility and moves towards its reservation value when approaching the negotiation deadline. Note that we do not assume a hard negotiation deadline, but work from the assumption that human negotiations typically last no longer than 7 to 8 rounds. Thus, it is assumed that an agent's tactics during a negotiation can be defined by a monotonically decreasing function. This model still allows that an opponent uses various kinds of tactics and no exact knowledge about an opponent's negotiation tactics is assumed. More specifically, the opponent's strategy is modelled as a probability distribution associated with a range of tactics (see Figure 4.3); as a result, each utility associated with an opponent's bid thus also has an associated probability.

In this paper we use linear functions to estimate the predicted utility value: $u'(b_t) = 1 - 0.05t$. This assumption allows us to compute the conditional probability $P(b_t|h_j)$ representing the probability of bid b_t given hypothesis h_j at time t . This is done by defining the probability distribution $P(b_t|h_j)$ over the predicted utility of b_t using the rationality assumption and the utility of b_t according to hypothesis h_j (see Figure 4.3). Here the predicted utility $u'(b_t)$ of a next bid of the opponent is estimated as $u'(b_{t-1}) - c(t)$ using a function $c(t)$ that is the most plausible model of the negotiation concession tactic used by the opponent. We use the following function to model the conditional distribution,

where $u(b_t|h_j)$ is the utility of bid b_t according to the hypothesis h_j :

$$P(b_t|h_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(u(b_t|h_j)-u'(b_t))^2}{2\sigma^2}} \quad (4.3)$$

This probability distribution can be used consecutively to update the probabilities of the hypotheses using Bayes' rule to compute $P(h_j|b_t)$.

The spread σ of the conditional distribution used in 4.3 defines the *certainty* of the agent about its opponent's negotiation tactics. If an agent is certain about the utility of an opponent's bid b_t then σ can be set to a low value. A higher level of certainty increases the learning speed, since hypotheses predicting an incorrect utility value of a bid in that case would get assigned an increasingly lower probability, and vice versa. Overestimating the level of certainty, however, may lead to incorrect results, and some care should be taken to assign the right value to σ .

4.4 Learning Approach

This section introduces the BOP algorithm. First, we explain the BOP learning algorithm. Second, we propose a computationally scalable solution for the BOP learning algorithm, called ScalableBOP. Finally, we explain how the BOP algorithm can be used to estimate the opponent's utility of a bid.

4.4.1 Bayesian Learning Algorithm for Opponent Preferences

The framework for learning introduced above can now be applied. In order to do so, the first step to perform is to initialize the probability distribution associated with each of the hypotheses in the hypothesis space H introduced in Section 4.3.1. This means either assigning a probability distribution to hypotheses based on available knowledge about opponent preferences, or, if no such *a priori* knowledge is available, to assign a uniform distribution.

During a negotiation at every time t when a new bid b_t is received from the opponent the probability of each hypothesis should be updated using Bayes' rule:

$$P(h_j|b_t) = \frac{P(h_j)P(b_t|h_j)}{\sum_{k=1}^m P(h_k)P(b_t|h_k)} \quad (4.4)$$

Here the conditional probability $P(b_t|h_j)$ represents the probability that bid b_t might have been proposed given hypothesis h_j (using the predicted utility according to the conditional

distribution (4.3)) and $P(h_j)$ is the current probability of hypothesis h_j . The normalization factor in the denominator of Bayes' rule ensures that the probability of the entire hypothesis space is 1.

The learning approach outlined will increase the probability of a hypothesis about an opponent's preference profile that is most consistent with the bid sequence received so far from that opponent and provides the best match with the utilities of these bids, estimated using the conditional probability distribution associated with tactics. As a result, the more consistent the predicted utility is with a hypothesis, the higher the probability associated with this hypothesis will be. It is possible that several hypotheses predict (almost) the same utilities for a given bid sequence, but this simply means that it is not possible to distinguish different preference profiles based upon that bid sequence and more evidence would be needed to do so.

The spread of the probability distribution $P(h_j)$ associated with the hypothesis space might also be used as a measure of the effectiveness of learning the opponent model. Presumably, successful learning of an opponent model will increase the probability of some of the hypotheses that best fit the bidding sequence received from an opponent and the number of hypotheses still considered viable would decrease. If not, the probability distributions $P(h_j)$ would remain a more or less uniform distribution. In the latter case the agent does not learn from the bids exchanged and it could use this fact in the negotiation strategy. For instance, negotiating against an erratic opponent that seems to more or less randomly propose bids, the agent might start using a Boulware strategy [4], in order to wait until an acceptable offer of the opponent is received.

Finally, during a negotiation an agent can use the updated probability distribution to compute estimates of the 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. The expected utility $\bar{u}(b_t)$ of a counteroffer b_t may be computed as follows, where w_i and e_i are the weights respectively evaluation functions predicted by hypothesis $h_j \in H$:

$$\bar{u}(b_t) = \sum_{j=1}^{|H|} P(h_j) \sum_{i=1}^n w_i e_i(x_i \in b_t) \quad (4.5)$$

4.4.2 Scalable Learning Algorithm

In this section, the learning approach is refined and an outline of a scalable algorithm (ScalableBOP) is discussed. Here our main concern will be the size of the hypothesis space $H = H^w \times H_1^e \times H_2^e \times \dots \times H_n^e$. This space is exponential in the number of issues and consists of $n!m^n$ hypotheses where m denotes the number of evaluation function hypotheses (see Figure 4.1). Clearly, even though the approach is very effective in small domains, it is not computationally feasible to update this many hypotheses in larger negotiation domains. In order to deal with larger domains, some additional independence assumptions will be introduced. As is to be expected, this will impact the performance

of the learning algorithm, but we will present additional experiments that show improved performance compared to that of the other strategies taken from the literature with which we compared our strategies.

To enable scaling of the proposed learning approach for negotiation domains of high dimensionality it will be assumed that the probability of individual components of a hypothesis $h = \langle h^w, h_1^e, \dots, h_n^e \rangle$ about a complete preference profile can be learned independently. That is, it will be assumed that weight ranking hypotheses h^w and the shape of each issue evaluation function h_i^e can be learned independently from each other. This is a reasonable approximation since each bid may be presumed to give at least some information about one issue relative to the available knowledge about the other issues.

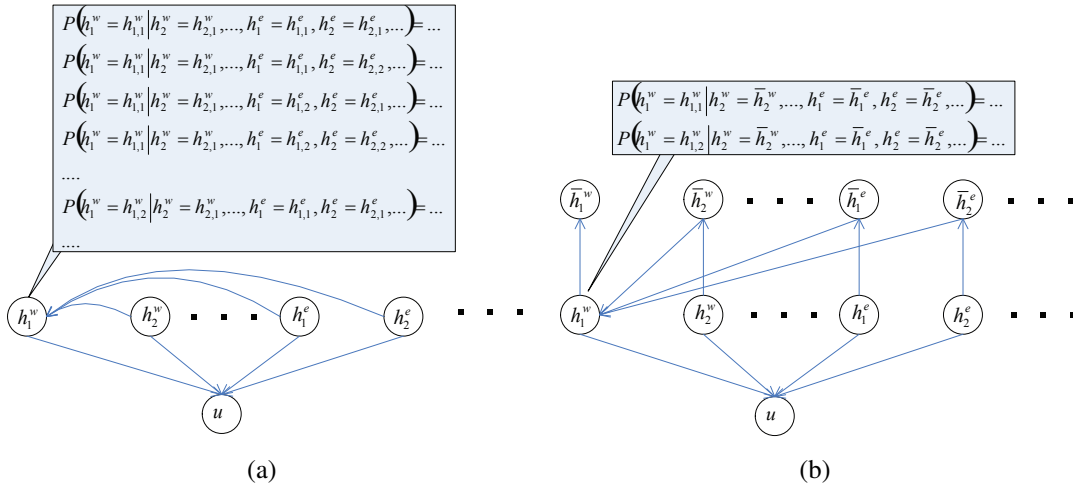


Figure 4.4: Bayesian network representing learning probabilities (a) over complete preference profiles hypotheses and over (b) individual hypotheses for weights and shapes of evaluation functions.

First, we will explain how each of the evaluation function hypotheses can be learned independently. The idea is illustrated in Figure 4.4.2. Figure 4.4.2(a) shows the approach outlined in Section 4.3 as a Bayesian network whereas Figure 4.4.2(b) illustrates how the independence assumption can be exploited to split up each hypothesis into its components and add these as nodes to the network. To simplify the notation we assume that symbol $h_{i,j}^e$ can be applied to a bid as a function and results in an evaluation value of the bid according to the evaluation function of hypotheses j for the issue i . The size of the local probability distribution table of each hypothesis in the original approach is $n!m^{n-1}$. In the approximation method, which introduces additional nodes for every hypothesis, the size of such a local probability distribution table is only m . Each of these additional nodes represents an expected value of the evaluation function for a given bid:

$$\bar{h}_i^e(b_t) = \sum_{j=1}^m P(h_{i,j}^e) h_{i,j}^e(b_t) \quad (4.6)$$

Second, we need to consider an approximation method for learning weight ranking hypotheses. Note that the number of possible weight orderings is $n!$ which is prohibitive for large n . To reduce the number of weight ranking hypotheses the normalization requirement associated with weights is relaxed. Instead of $n!$ hypotheses a set of m hypotheses for each weight is introduced, where each hypothesis represents a possible value of the weight. Similar to the hypotheses for evaluation functions we introduce the symbol $h_{i,j}^w$ to denote the hypothesis about the value of the weight for issue i according to hypothesis j , and will also sometimes use it to denote the value of the associated weight, i.e., $h_{1,1}^w = 0, h_{1,2}^w = 0.1, h_{1,3}^w = 0.2, \dots$. Then, the expected value of an issue weight can be calculated as follows:

$$\bar{h}_i^w = \sum_{j=1}^m P(h_{i,j}^w) h_{i,j}^w; \quad (4.7)$$

The nodes of expected values for evaluation functions and weights are used to update local probability distributions only. The expected utility of a bid b_t is now calculated as follows:

$$\bar{u}(b_t) = \sum_{i=1}^n \bar{h}_i^w \bar{h}_{i,j}^e(b_t) \quad (4.8)$$

Since a utility function is assumed to be linearly additive this approximation of weight ranking hypotheses does not influence the selection of a bid that maximizes the opponent's utility (when computing a counteroffer). However, the approximation may affect the prediction of the utility of an opponent's bid thus influencing the quality of learning when updating the probability of the hypotheses in line with the conditional distribution associated with the opponent's tactics.

Now we proceed and show that this approximation solves the scalability problem. Note, that instead of normalizing probabilities over complete set of possible utility spaces the probability distribution over weights and evaluation functions are normalized for every issue:

$$\sum_{j=1}^m P(h_{i,j}^w) = 1, i = 1, \dots, n; \sum_{j=1}^m P(h_{i,j}^e) = 1, i = 1, \dots, n; \quad (4.9)$$

Taking this into account, we can show that the expected utility of a bid is the same as in the original approach when the same a priori probability distributions are used. The main idea concerns the modification of the learning itself, i.e., the update of the probabilities associated with hypotheses about single weights and evaluation functions of single issues. Instead of calculating the probability distribution for a given hypothesis with respect to all possible partial opponent models we now use the best prediction (or expected value) of the *current* model. In other words, the probability distribution of a hypothesis is estimated

by using the probability distributions provided by the model learned so far. The update of the probability of a hypothesis thus assumes that these probability distributions of other hypotheses yield a reasonably good prediction of the opponent's preferences.

It can be shown that if this is the case, the obtained probabilistic model would correspond to the same model built for the hypothesis space over complete preference profiles. In other words, we can show for $h_k \in H$ that:

$$P(h_k) \leftarrow \prod_{i=1}^n P(h_{i,j}^w \in h_k) \prod_{i=1}^n P(h_{i,j}^e \in h_k), h_k \in H \quad (4.10)$$

It thus is clear that the approach will greatly benefit from the use of partial domain knowledge when available. In that case, the update of the probability distribution associated with a hypothesis would not be based on probabilistic information associated with the opponent model but on given domain knowledge.

4.4.3 Updating Probabilities of Hypotheses

Because the first bid has maximal utility for a negotiator according to one of the rationality assumptions introduced earlier, this bid does not provide any information about an opponent's issue priorities. The first bid thus only can be used to update probability distributions of hypotheses about an opponent's evaluation functions and the probability distributions of hypotheses about weights can be updated only after the agent has received more than one bid from an opponent.

Taking this into account, the conditional distribution associated with tactics can be used to update the hypothesis of issue k using the expected evaluation values and weights of the rest of the issues as defined by the current opponent model. So, suppose we need to update the probability distribution of the hypothesis for issue k after receiving a bid b_t from the opponent. In order to do so, we introduce a partial expected utility $\bar{u}_{\langle -k \rangle}(b_t)$ of bid b_t that does not take the contribution of issue k to the utility of the bid into account, and is defined as follows:

$$\bar{u}_{\langle -k \rangle}(b_t) = \sum_{i=1,2,\dots,k-1,k+1,\dots,n} \bar{h}_i^w \bar{h}_{i,j}^e(b_t) \quad (4.11)$$

The probability of the hypotheses over the shape of the evaluation function can then be updated according to Bayes' rule as follows:

$$P(h_{k,j}^e | b_t) = \frac{P(h_{k,j}^e) P(\bar{u}_{\langle -k \rangle}(b_t) + h_{k,j}^e \bar{h}_k^w | h_{k,j}^e)}{\sum_{i=1}^m P(h_{k,i}^e) P(\bar{u}_{\langle -k \rangle}(b_t) + h_{k,i}^e \bar{h}_k^w | h_{k,i}^e)}, j = 1, \dots, m \quad (4.12)$$

where $\bar{h}_k^w$ is the expected value of the weight of issue k .

The probability of the hypotheses related to the weight of issue k can be updated in a similar way as follows:

$$P(h_{k,j}^w | b_t) = \frac{P(h_{k,j}^w)P(\bar{u}_{\langle -k \rangle}(b_t) + h_{k,j}^w \bar{h}_k^w | h_{k,j}^w)}{\sum_{i=1}^m P(h_{k,i}^w)P(\bar{u}_{\langle -k \rangle}(b_t) + h_{k,i}^w \bar{h}_k^w | h_{k,i}^w)}, j = 1, \dots, m \quad (4.13)$$

Because the application of Bayes' rule to multiple hypotheses needs to be implemented as a sequential procedure, care should be taken to perform a Bayesian update by using the expected utility, weights and evaluation values that are derived from the probability distribution before any Bayesian update has been performed. Otherwise, any hypotheses that are updated after other hypotheses have been updated would be biased by the updated probability distributions of these hypotheses that already have been updated. Additionally distributions of a priori probabilities have to be adjusted in such a way that the sum of the expected values of the weights equals one, i.e.:

$$\sum_{i=1}^n \bar{h}_i^w = 1 \quad (4.14)$$

4.5 Quality Assessment Method

The quality assessment method we propose has three components: (i) quality measures to estimate the learning performance, (ii) criteria for selecting a diverse range of negotiation domains and preference profiles on these domains, and (iii) criteria for selecting a number of negotiation strategies of the opponent. These components then are used to define an experimental setup to obtain data to analyze learning quality by means of a negotiation tournament.

The first component consists of several similarity measures that provide a metric for assessing the accuracy of the learned preference profile with respect to the actual preference profile. We discuss several measures that can be used to assess the quality of the learned preference profile. Apart from the restriction on utility functions which need to be linearly additive, the second component of the method consists of several additional criteria for selecting negotiation domains such as size and complexity of the domain, and the similarity of the preference profiles of the negotiating parties. These criteria are used to define the experimental setup of the negotiation tournament. The third component provides criteria for selecting negotiation strategies that should be used by negotiating agents in the tournament. Since learning of an opponent's preference profile in single-instance negotiations has to be accomplished with only the observations of the opponent's negotiation moves, typically such learning algorithms use assumptions about an opponent's behaviour. Although this assumption is reasonable and can be applied in typical negotiation settings, it is important to assess the robustness of a learning technique also when negotiating against agents that use strategies that do not comply with this assumption. It thus is important to

incorporate a diverse range of negotiation strategies in any experimental setup to evaluate learning quality.

4.5.1 Quality Measures

In this Section we discuss two quality measures to assess learning quality that are based on two metrics to measure the distance between the actual preference profile of an opponent and the learned preference profile. These quality measures are applied to both the complete preference profiles or utility functions, as well as to the issue priorities or weights.

The learning task of learning an opponent's preference profile clearly is an approximation problem. The task is to re-constructs the actual utility function u of the opponent by means of a learning technique resulting in an approximate function $\tilde{u}$. A quality measure with respect to learning preference profiles therefore can be defined as a distance metric of two utility functions, and can be formally represented as $d(u, \tilde{u})$.

Ideally, the approximation $\tilde{u}$ of an opponent's utility function would provide an accurate prediction of the exact utility value an opponent associates with an outcome. Some strategies like the Tit-for-Tat-based strategy introduced in [4] depend on the accuracy of cardinal values of the utility function of the opponent since a negotiation move is chosen based on an estimate of the concession the other party made in the previous move. It therefore is important to have a distance metric that can be used to measure the accuracy of the cardinal values predicted by the learned profile. Here we use Pearson's correlation coefficient for that purpose. This coefficient represents the degree of linear relationship between two variables and is defined as follows:

$$d_{pearson}(u, \tilde{u}) = \frac{\sum_{\omega \in \Omega} (u(\omega) - \langle u \rangle)(\tilde{u}(\omega) - \langle \tilde{u} \rangle)}{\sqrt{\sum_{\omega \in \Omega} (u(\omega) - \langle u \rangle)^2 \sum_{\omega \in \Omega} (\tilde{u}(\omega) - \langle \tilde{u} \rangle)^2}} \quad (4.15)$$

where $\langle u \rangle$ (respectively $\langle \tilde{u} \rangle$) denotes the average utility over the outcome space defined by utility function u ($\tilde{u}$). The Pearson's correlation coefficient takes a real value from the interval $[-1; 1]$. A value of $+1$ means that there is a perfect positive linear relationship between variables, whereas a value of -1 means that there is a perfect negative linear relationship between variables. A value of 0 means that there is no linear relationship between the two variables.

Although a perfect match of cardinal values of the actual and learned utility function would be ideal, in practice it may be sufficient and more important to approximate the preference ranking of outcomes by an opponent (cf. [5]). For example, negotiation strategies that aim at maximizing an opponent's utility by means of walking on a utility iso-curve in one's own preference profile only need adequate information about an opponent's

ranking of outcomes. It is sufficient when using such strategies to possess accurate ordinal ranking information.

To estimate the distance between the rankings of the bids given the actual utility function of the opponent and the learned utility function, a metric is introduced that compares all outcomes in the outcome space pairwise. In order to do so, a ranking relation $\prec_u$ is defined as follows: $\forall \omega_i, \omega_j \in \Omega, \omega_i \prec_u \omega_j \Leftrightarrow u(\omega_i) < u(\omega_j)$. Using this ranking relation, we can define a conflict indicator function adapted from [7] to measure conflicting rankings given arbitrary utility functions u and $\tilde{u}$. The conflict indicator function is defined as follows:

$$c_{\prec_u, \prec_{\tilde{u}}}(\omega_i, \omega_j) = \begin{cases} 1 & \text{if } (\omega_i \preceq_u \omega_j \wedge \omega_j \prec_{\tilde{u}} \omega_i) \vee (\omega_i \prec_u \omega_j \wedge \omega_j \preceq_{\tilde{u}} \omega_i) \\ & \vee (\omega_i \preceq_{\tilde{u}} \omega_j \wedge \omega_j \prec_u \omega_i) \vee (\omega_i \prec_{\tilde{u}} \omega_j \wedge \omega_j \preceq_u \omega_i), \\ 0 & \text{otherwise.} \end{cases} \quad (4.16)$$

The conflict indicator function yields 1 when the ranking relation of two arbitrary outcomes ω, ω' based on the learned utility space $\tilde{u}$ is not the same as the ranking relation based on the actual utility space of the opponent u ; if the rankings based on both utility functions match the conflict indicator takes the value of 0.

Using this ranking relation, we can define a conflict indicator function adapted from [7] to measure conflicting rankings given arbitrary utility functions u and $\tilde{u}$. The conflict indicator function is defined as follows:

$$d_{\text{ranking}}(u, \tilde{u}) = \frac{1}{|\Omega|^2} \sum_{\omega \in \Omega, \omega' \in \Omega} c_{\prec_u, \prec_{\tilde{u}}}(\omega, \omega') \quad (4.17)$$

In [7] various properties of this distance measure are proved, including e.g. reflexivity, symmetry and the triangle inequality property.

It is useful to not only apply the distance measures to complete preference profiles but also to apply it to the issue priorities or weights in such a profile. In Section 4.6 we apply the assessment method to the BOP algorithm. In this learning approach the different components of a linearly additive utility function, i.e. weights and evaluation functions, are learned in a different way. In order to obtain experimental data about these different learning processes we therefore also define similar distance measures to those discussed above for measuring distance of actual and learned issues weights.

The set of weights can be represented as a weight vector, and it is not hard to define the Pearson correlation coefficient for the vectors of weights. The coefficient is defined as follows:

$$d_{\text{pearson}}(W, \tilde{W}) = \frac{\sum_{i=1}^n (w_i - \langle w \rangle)(\tilde{w}_i - \langle \tilde{w} \rangle)}{\sqrt{\sum_{i=1}^n (w_i - \langle w \rangle)^2 \sum_{i=1}^n (\tilde{w}_i - \langle \tilde{w} \rangle)^2}} \quad (4.18)$$

To calculate the ranking distance between the two weight vectors W and $\tilde{W}$ a ranking relation is constructed on the weights of the corresponding vector as follows: $i = 1 \dots n, j = 1 \dots n, i \prec j \Leftrightarrow w(i) < w(j)$, where $w(i) = w_i$. Then, the conflict indicator $c_{\prec_W, \prec_{\tilde{W}}}(i, j)$ can be defined in the same way as for utility functions. The ranking distance of two weight vectors is defined as follows:

$$d_{\text{ranking}}(W, \tilde{W}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n c_{\prec_W, \prec_{\tilde{W}}}(i, j) \quad (4.19)$$

4.5.2 Negotiation Domains and Profiles

Whereas precise mathematical metrics can be defined for measuring distance of preference profiles, for the selection of an adequate set of domains to be used in the experimental setup less formal criteria are proposed here. The main reason is that it is impossible to assess a learning technique on the space of all negotiation domains and associated preference profile. Ideally, then, one would use an experimental setup based on random sampling of the domains and profiles in order to deal with this problem. However, it is not clear how to setup such a sampling procedure. As an example, we found that the predictability of issue preferences (see below) may influence the outcomes of negotiation strategies. It is not particularly clear, however, how to obtain a random sample which would be an adequate representation of domains with and without predictable issues. Instead, we therefore discuss and propose to use three factors for selecting domains that are relevant in testing the learning quality.

Size of the negotiation domain. The amount of information exchanged during the negotiation is limited in a closed negotiation since we can rely only on observed negotiation moves of an opponent, which affects learning quality. The amount of information needed by a learning technique typically depends on the model structure and the size of the parameter space that is to be learned. Therefore, a learning technique has to be assessed on negotiation domains of various sizes and of various complexity. Since in any negotiation the number of issues is one of the most important factors that determine the complexity of the preferences profile, a set of domains should be selected that range from a low number of issues to higher number of issues.

Predictability of the preferences. Most learning techniques for learning an opponent's preference profile use assumptions about the structure of the preference profile (e.g. see [3, 22]). Among others such techniques may rely on the predictability of issue preferences [8]. Issues are called predictable when even though the actual evaluation function for the issue is unknown, it is possible to guess some of its global properties. For example, a price issue typically is rather predictable, where more is better for the seller, and less is better for the buyer, and the normal ordering of the real numbers is maintained; an

issue concerning colour, however, is typically less predictable. Learning even ranking preferences related to issue values of unpredictable issues therefore is more difficult.

The set of selected negotiation domains for any experimental setup therefore ideally should consist of a balanced mix of predictable and unpredictable issues. In principle, the higher the number of unpredictable issues the more complicated the learning of a corresponding profile becomes.

Opposition of preferences. The results of analyzing negotiation dynamics presented in [8] revealed that some negotiation strategies are sensitive to preference profiles with compatible issues. Issues are compatible if the issue preferences of both negotiating parties are such that they both prefer the same alternatives for the given issue. Negotiation strategies may more or less depend on whether preferences of the negotiating parties are opposed or not on every issue. That is, using some strategies it is harder or even impossible to exploit such common ground and agree on the most preferred option by both parties for compatible issues (humans are reported to have difficulty with this as well; cf. [20]). A selection of preference profiles should therefore take into account that both preference profiles with and without compatible issues are included.

The notion of opposition can be made more precise. Conceptually, it represents a degree of conflict of interests between the parties. In other words, there is a conflict of interests if one party prefers outcome ω over outcome ω' and the other party prefers outcome ω' over outcome ω . In [11] a notion of local opposition based on the gradients of the utility functions of both parties is defined for each outcome in the negotiation domain. Intuitively, if the gradients point to opposite directions then the preferences of the negotiation parties are opposed. The more colinear the gradients are the closer (more compatible) the preferences of the parties. Although it is possible to generalize the notion of local opposition relative to an outcome to a more global notion of opposition of utility functions, we propose to reuse the distance measures for preference profiles to measure the level of opposition present.

Negotiation Strategies of the Opponent

The results of the analysis presented in [8] also have shown that the performance of a negotiation strategy can be significantly influenced by the negotiation strategy of the opponent. For example, the class of pure time-dependent tactics (TDT; see [4]) does not take into account the negotiation moves of opponents and selects the next offer to propose in a negotiation based on how close one is to the negotiation deadline. Whereas TDT tactics are insensitive to opponent moves, negotiation strategies in the class of behaviour-dependent tactics (BDT) do base their choice of offer on the offers received so far from the opponent. A variety of strategies therefore is needed to assess the quality of learning, which includes strategies that belong to the TDT class, the BDT class as well as mixes thereof.

The selection of strategies to be used in an experimental setup should be able to test the robustness of the learning technique with respect to various opponents that use different types of negotiation strategies. For example, to enable learning of opponent preferences from the observed negotiation moves (offers) in BOP a model of opponent strategy is used (see also [1, 22]). Such rationality assumptions might however be exploited and it should be tested if a learning technique is robust against strategies like the Zero-Intelligence strategy that uses an irrational random tactic [6].

Again we do not claim to present an exhaustive coverage of the criteria discussed, but present a selection to illustrate. The following negotiation strategies have been used by the negotiating parties in our experimental setup:

The *Time-Dependent Tactic* (TDT) strategy [4] proposes as a next offer a bid that has a decreased utility compared to the previously proposed offer. TDT does not use any information about the opponent. Utility of the next offer is decreased according to a concession function, which can be linear (fixed concession size), bouldware (make small concession in the beginning of a negotiation and big ones at the end), and conceder (make big concession in the beginning of a negotiation and small ones at the end).

The *ABMP* strategy from [10] which is a concession oriented approach in the TDT class, and takes no heed of knowledge about the domain or the opponent. The ABMP strategy uses a non-linear concession tactic. It concedes more in the beginning of the negotiation when the gap between the negotiation positions is big and decreases the size of the concession when the negotiation positions approach each other. As such, it is an example of a so-called conceder tactic (cf. [4]).

The *Trade-off Strategy*, taken from [5], uses so-called similarity criteria and exploits domain knowledge. The Trade-off strategy is an example of a Behaviour-dependent strategy. In our experiments we allowed three smart steps and a concession of 0.05 for the smart meta strategy.

Zero-Intelligence, taken from [6], is a random strategy that makes random jumps through the outcome space. The ZI agent used a reservation point of 0.6 in our experiments to avoid making offers that have a ridiculously low utility. The ZI strategy plays a role as a baseline strategy.

4.6 Experimental Setup

According to the methodology proposed in this section a set of negotiation domains and preference profiles was created. The set of the domains and preference profiles used by the opponent of the BOP algorithm was designed in such a way that the number of negotiations to be run would be minimal but still have a sufficient variation over the following characteristics: the size of the domain, number of predictable issues, and opposition of the preferences. In this section we describe the procedure that was used to generate the experimental set of the negotiation domains and preference profiles.

The number of issues in a negotiation domain was used to vary the size of the negotiation domains. The number from 4, 6, 8, to 10 issues. All issues were discrete and had 10 values per issues. The BOP algorithm is able to deal with domains up to 7 issues in reasonable time. Therefore, both the BOP and the ScalableBOP algorithms were tested on the domains with 4 and 6 issues to see the effect of decision choices made in the ScalableBOP algorithm on the learning performance. For the domains with 8 and 10 issues the ScalableBOP algorithm was used. The biggest negotiation domain in the experimental domain has 10^{10} possible agreements which is bigger than any real negotiation domain presented in the literature (see e.g. [19] for the “AMPO vs. City” domain that has only 7,000,000 possible agreements).

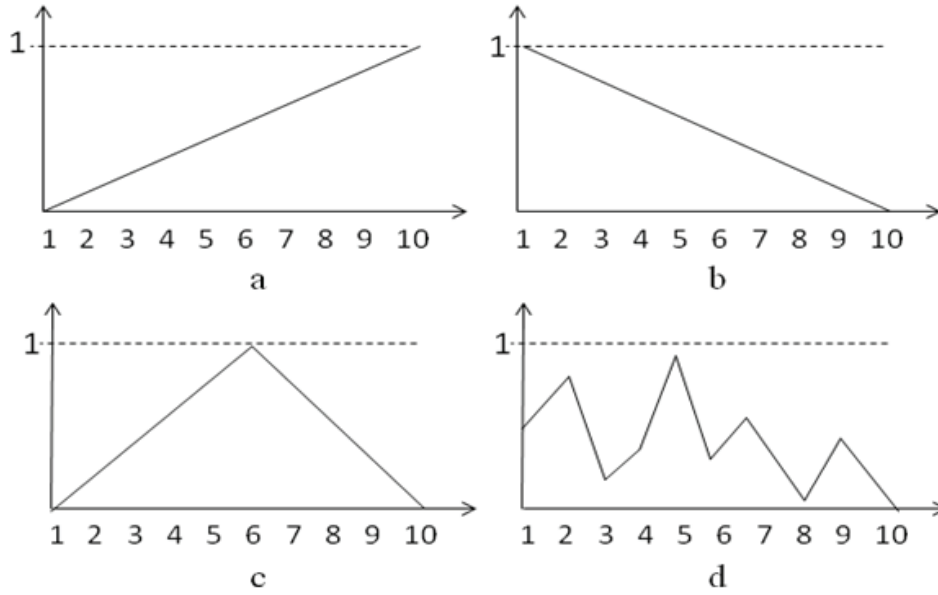


Figure 4.5: Examples of evaluation functions used in the experimental setup: a - “uphill”, b - “downhill”, c - “triangle”, d - “unpredictable”.

To model preferences over values of individual issues two types of issue were made: *predictable* and *unpredictable*. For the predictable issues one of the three types (*uphill*, *downhill*, and *triangle*) of an evaluation function was used. Similarly, for unpredictable issues, three types of evaluation function were generated by means of assigning a random number as an evaluation value for every alternative of the issue. Figure 4.5 shows examples of the evaluation functions for the predictable and unpredictable issues. The evaluation functions were combined with one of two possible weight vectors. The first vector represents weights that linearly decay from the most important issues to the least important issue, e.g., for a 4-issues domain the vector could be $\langle 0.4; 0.3; 0.2; 0.1 \rangle$. In the second vector the weights decay from the most important issues to the least important issue exponentially, e.g., for a 4-issues domain the vector could be $\langle 0.53; 0.27; 0.13; 0.07 \rangle$. Permutations of the weights in the vector do not influence the learning performance of BOP and, therefore, do not need to be included in the experimental setup.

To have a sufficient variation of the preference profiles in our experimental setup without exploding the number of negotiations needed to be run we produced all possible permutations of the evaluation functions for the two most important issues in a domain according to the weights. The rest of the issues in total have significantly lower influence on the utility and, therefore, were assigned an evaluation function randomly. The preference profile used by the agent that uses the BOP algorithm does influence the quality of learning of the opponent’s profile. However, variation of the preference profiles used by the BOP agent is required to study the influence of the opposition of preferences on the learning. To control the number of negotiations we created eight types of profiles for which we varied the following:

1. two types of evaluation functions: all “downhills” and all “uphills”;
2. two weights vectors: with linear and exponential decay of the weights;
3. every weights vector was reordered: e.g. $\langle 0.4; 0.3; 0.2; 0.1 \rangle$ would become $\langle 0.1; 0.2; 0.3; 0.4 \rangle$.

To show the effectiveness of our approach to learn the opponent model and to use it to find a good counteroffer the BOP learning technique was combined with a simple strategy and integrated in a negotiating agent. The strategy used by the BOP agent is based on the smart meta-strategy of [5]. The agent starts with proposing a bid that has maximal utility given its own preferences. Each 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.

The basic idea of the smart meta-strategy is to propose a counter-offer that has the same utility (lies on the same utility iso-curve) as the previous bid of the agent but improves the utility of the opponent whenever possible. Formally, the strategy searches for a bid b_{t+1} that satisfies the following formula, in which u_{own} denotes the agent’s own utility function and τ denotes a target utility:

$$b_{t+1} = \arg \max_{b \in \{x \mid |u_{own}(x) - \tau| \leq \delta\}} \bar{u}(b) \quad (4.20)$$

The set $\{x \mid |u_{own}(x) - \tau| \leq \delta\}$ represents the utility iso-curve of bids that have the same utility for the agent, (within a small interval $[\tau - \delta; \tau + \delta]$) but might have different utilities for its opponent. The strategy selects a bid from the iso-curve that maximizes the expected utility of the opponent. The bid b_{t+1} lies on the predicted Pareto frontier according to the current opponent model. If it is not possible to find a bid that thus improves the utility of the opponent, a concession step will be performed after performing smart steps (i.e. steps that stay on the same iso-curve and try to improve the next bid for the opponent by using the updated opponent model). The agents perform a concession step by decreasing the target utility τ of their next bid by a fixed concession step c .

In all experiments after every update of a preferences model the ranking and the Pearson’s quality measures were estimated as explained in Section 4.5.1. To measure efficiency of

the counteroffers generated by the BOP agents a distance to the Pareto frontier was calculated. The distance was calculated as a difference in utility according to the opponent's utility function between the counteroffer generated by the BOP agent and the corresponding offer on the Pareto frontier lying on the same iso-level in the opponent's utility space.

In addition, to illustrate negotiation performance of the BOP learning algorithm as well as the BOP agent on more realistic domains two sets of experiments were run using real negotiation domains described in the negotiation literature: one based on a negotiation domain with 4 issues taken from [5], and one based on a negotiation domain with 10 issues taken from [19]. To compare the performance of the Bayesian learning approach, the agents using opponent modelling were compared with agents using the Trade-off strategy and the ABMP strategy. Two variants of BOP agents were tested: one with and one without initial domain knowledge; the first to compare with the Trade-off strategy which uses domain knowledge and the second to compare with the ABMP strategy which does not. All agents played against the same opponent, which used the Trade-off strategy, to be able to compare negotiation traces and results.

4.7 Experimental Results

First of all we check whether the BOP algorithm is able to learn in small domains. Figure 4.6 shows learning curves for the 4 and 6 issues domains averaged over all negotiation domains. The horizontal axis represents negotiation rounds. Round 0 corresponds to the initial state of the opponent model when it has not been updated yet and the distance measures correspond to a model with uniform probability distribution assigned to the hypothesis space. The initial ranking distance equals 0.5 meaning that at the beginning in the current opponent model half of the rankings between outcomes drawn from the complete outcome space is correct.

As expected, in all cases the ranking distance between the learned model and the original opponent's preference significantly decreases with every negotiation round. The BOP algorithm learns the preferences of the opponent best when the opponent uses the TDT strategy. In our experimental setup the TDT strategy uses linear concession tactics, which is more consistent with the opponent tactics design choice made in the BOP algorithm (see Section 4.3.2). The Trade-Off strategy uses semi-linear concession tactics. As a result, the learning performance of the BOP algorithms in negotiations against Trade-Off strategy is similar to the negotiations where opponent uses the TDT strategy. The ABMP strategy uses non-linear concession tactics that explains a slightly worse learning performance compared to the TDT and the Trade-Off strategy. The learning measure is not monotonous for the ABMP strategy and increases to some extent at the end of the negotiation due to accumulation of the difference between the size of the concessions according to the opponent's strategy model used by BOP and the actual concessions made by ABMP. As expected, the BOP algorithm learns the opponent preferences slower in case of the ZI negotiation strategy of the opponent. However, it is still capable of learning the

opponent's preferences to some extent.

The results for the Pearson distance show that the BOP algorithm is capable of learning the cardinal values of the utility function. Note, that in general the Pearson distance is correlated with ranking distance for the proposed learning algorithm. This can be explained by the nature of the hypothesis space of the learning algorithm. The algorithm calculates opponent's utility values of a bid as an expected value of a random variable. The expected value is a sum of the utilities according to the hypothesis weighted according to their probabilities. Thus, even if the more detailed structure of the opponent's preferences is not learned by the agent the information learned can still be used to approximate the utility function of the opponent as a linear combination of the set of all hypotheses. For some strategies like the relative Tit-for-Tat-based strategy introduced in [4] it is important to accurately estimate cardinal values of the utility function of the opponent from the opponent's previous moves to calculate the agent's own next move.

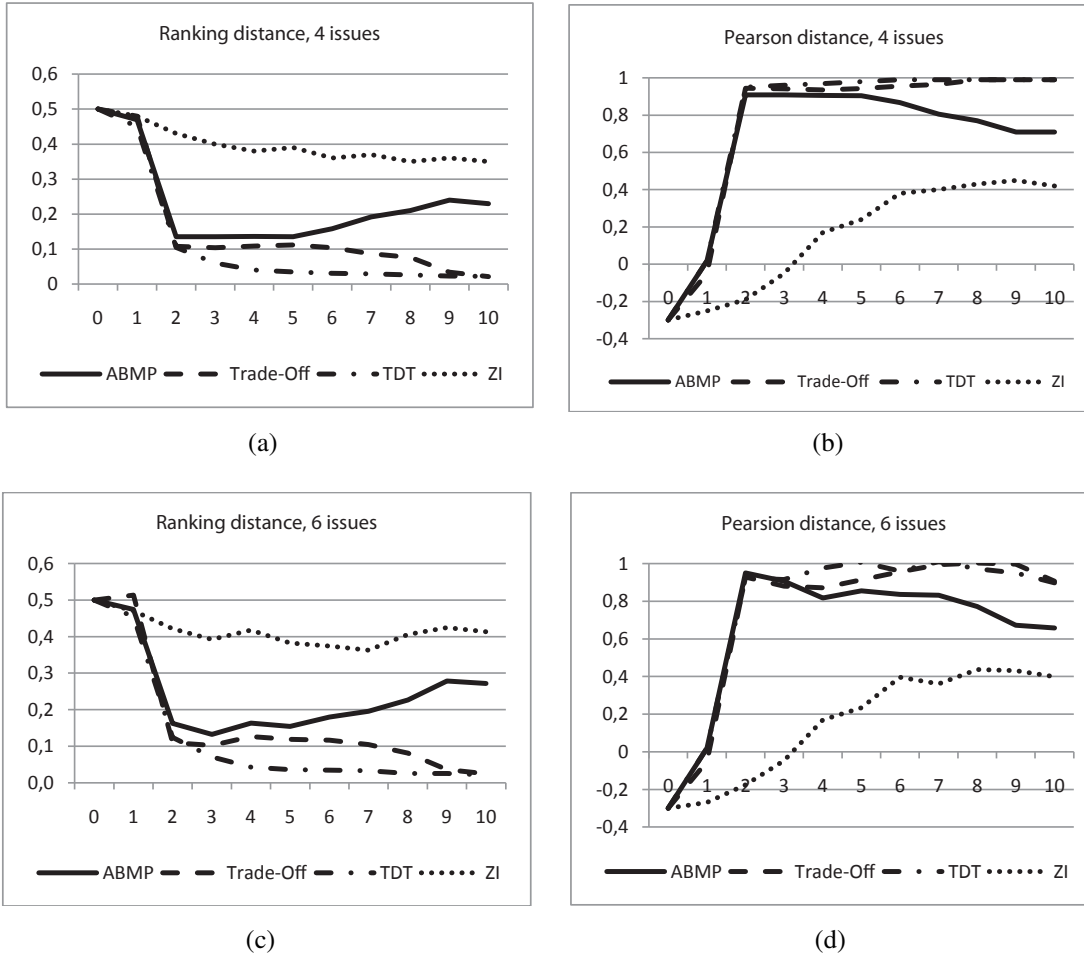


Figure 4.6: Learning quality measures for BOP algorithm on the 4 issues (a),(b) and 6 issues (c), (d) domains

In general, the learning quality is better in smaller negotiation domains. This follows from

a comparison of the BOP algorithm results on the 4 and 6 issues domains (see Figure 4.6). The same holds for the ScalableBOP algorithm (see Figure 4.7,4.8). Furthermore, the slope of the learning curves is less steep meaning that it takes longer to learn a model. This is explained by the fact that the bigger domain the more variables the algorithm has to learn. Comparison of the BOP and ScalableBOP learning performance on the 4 and 6 issues domains shows that the design choices made in ScalableBOP to decrease computational complexity have a rather small impact on the accuracy of the learned model.

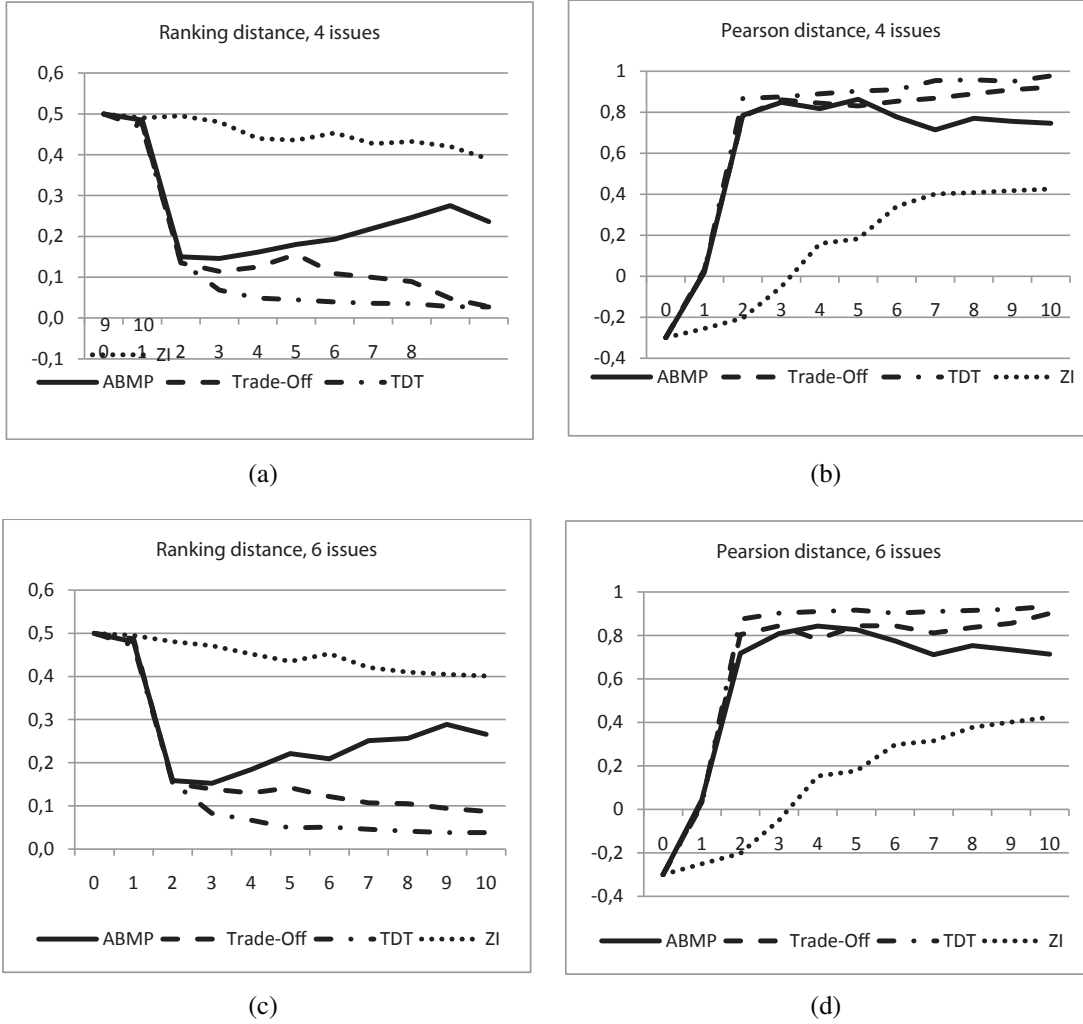


Figure 4.7: Learning quality measures for ScalableBOP algorithm on the 4 issues (a),(b), 6 issues (c), (d) domains

An important source for the “win-win” outcomes is the making of trade-offs between less and more important issues. Therefore, to be useful the ScalableBOP learning algorithm must be able to learn weights of the issues in opponent’s preference profile. Figure 4.9 shows learning quality measures for the weights (see Section 4.5.1 for details on the quality measures for weights). The Pearson distance on Figure 4.9 shows that the ScalableBOP algorithm is able to learn ranking of the weights as well as the cardinal values

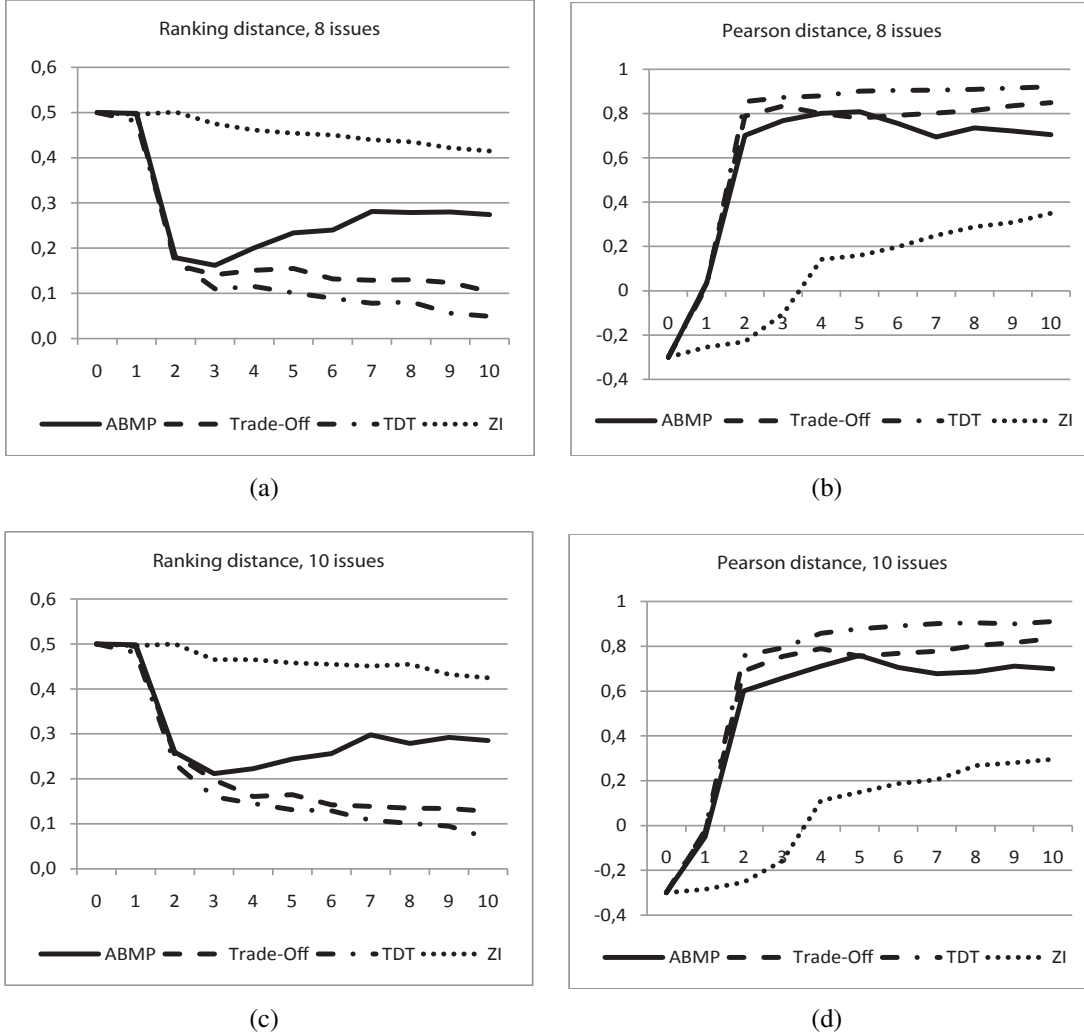


Figure 4.8: Learning quality measures for ScaleableBOP algorithm on the 8 issues (a), (b), and 10 issues (c), (d) domains

of the weights. The more precise the absolute values of the weights the more accurate the trade-offs between the issues are. Exact trade-offs allow a strategy to get closer to the Pareto frontier in its offers. Apparently, BOP is an algorithm that enables strategies to do so.

According to our requirement the BOP algorithm should not depend on the level of opposition of the preferences of the negotiating parties. Figure 4.10 shows the relationship between the quality of learning (vertical axis) and the opposition of the preferences (horizontal axis) on the 4 and 6 issues domain. Every point represents the quality of learning after 3rd round in a single negotiation session. From this we conclude that the distribution of the points does not correlate to the opposition of the preferences. The level of the clusters of the points depends only on the opponent's strategy. The points are concentrated mostly in the middle of the horizontal axis only due to the way the preference profiles

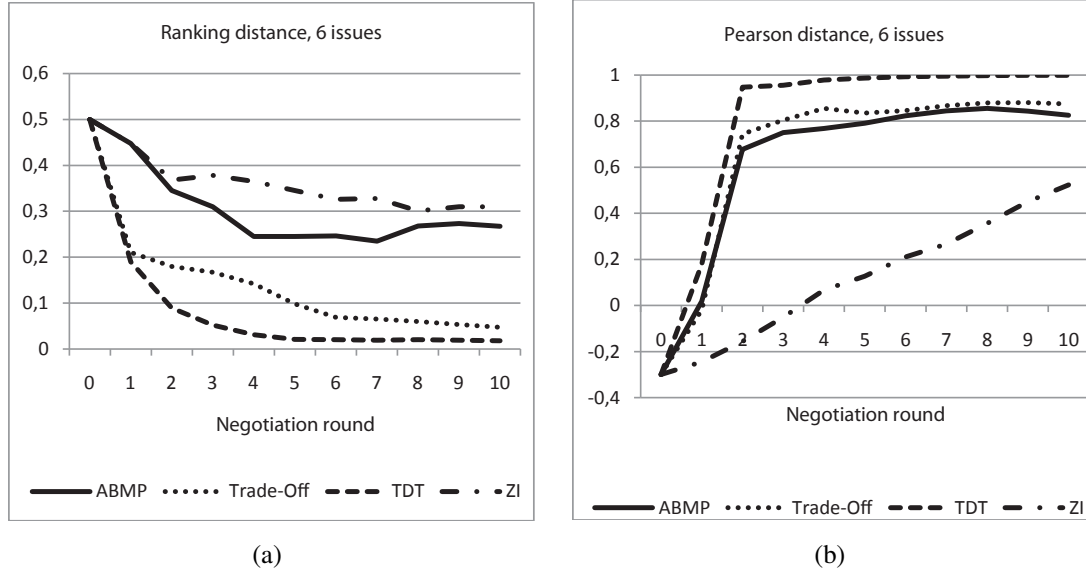


Figure 4.9: Learning quality measures for weights for ScalableBOP algorithm on the 6 issues domain

were generated in the experimental setup.

Another interesting observation is that the learning algorithm approximates the absolute, cardinal values of the utility function and weights quite well in domains with unpredictable issues (see Figure 4.11). This can be explained by the nature of the hypothesis space of the learning algorithm. The algorithm calculates opponent's utility values of a bid as expected values of a random variable. The expected value is a sum of the utilities according to the hypothesis weighted according to their probabilities. Thus, even if the more detailed structure of the unpredictable issues is not learned by BOP the information learned can still be used to approximate the utility function of the opponent as a linear combination of the set of all hypotheses.

The BOP as well as ScalableBOP algorithms can be initialized with a priori knowledge. Such knowledge can be derived, for instance, from the negotiation domain knowledge, e.g., a buyer would prefer to minimize the price issue or a seller would give the price issue the highest priority over all other issues. This knowledge can be used to initialize certain hypotheses with higher a priori probabilities. Non-uniform initial probability distributions where the hypotheses that are most likely to be correct have higher probability would increase the learning speed and accuracy. Figure 4.12 shows the results of the experiments in which the BOP algorithms were initialized with domain knowledge. Observe that the initial distance between the learned opponent model and the original opponent's preferences is smaller than in cases with uniform initial probability distributions. This means that already in the early rounds of negotiation the model of the opponent preferences has high quality. Furthermore, the ScalableBOP algorithm converges to a smaller distance when domain knowledge is used.

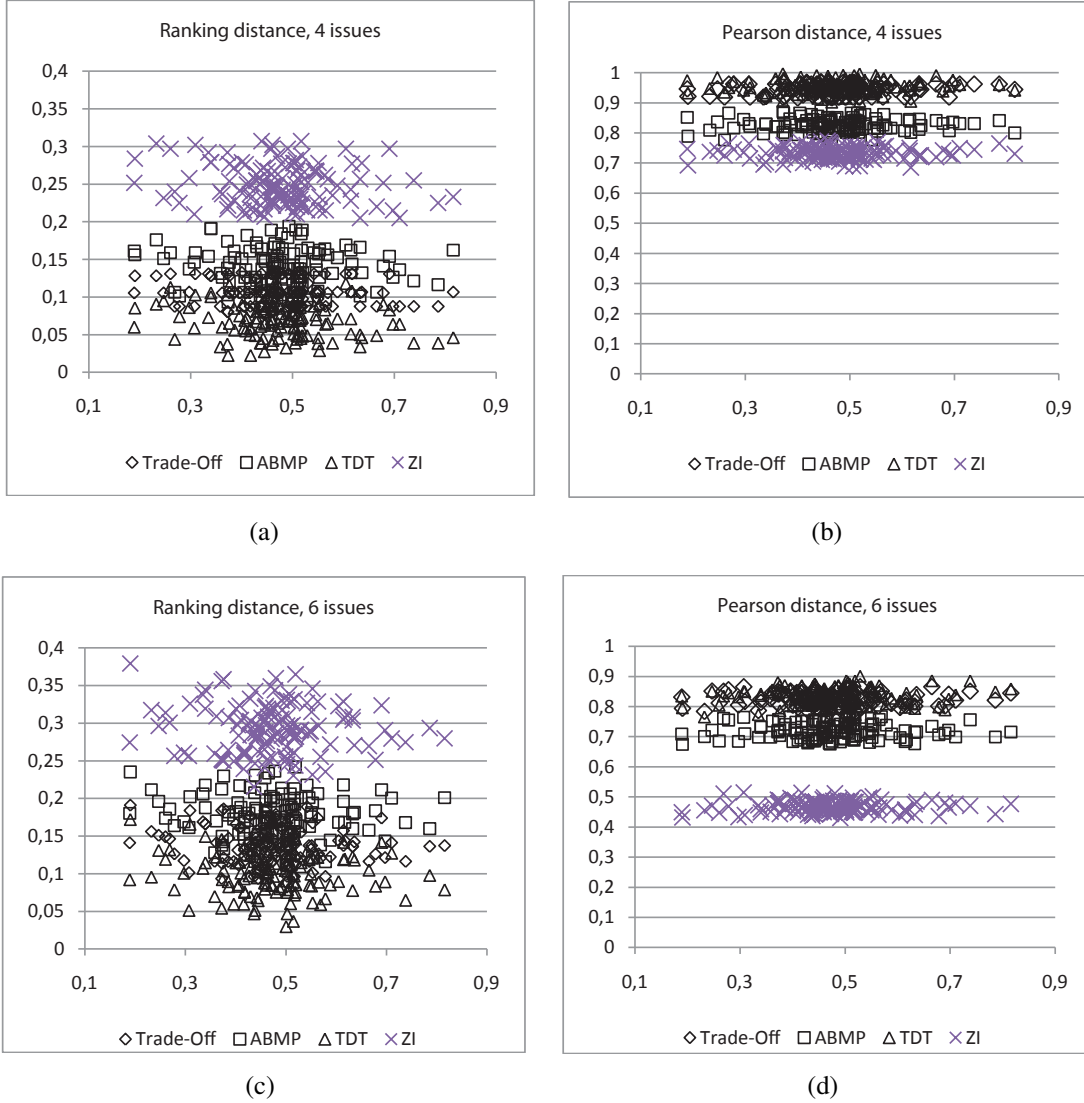


Figure 4.10: Distribution of the learning quality measures for the 3rd round.

Figure 4.13 shows learning quality curves for the BOP algorithm in the Employee-Employer domain and the AMPO vs City domain. The results obtained on these domains are very similar to that of the experimental setup with generated preference profiles. The learning quality is better in the Employee-Employer domains due to high predictability of all issue and smaller size when compared to the AMPO vs City domain. Despite the fact that the AMPO vs City domain is large and has few unpredictable issues the learning quality is still good.

To see the quality of the counteroffers generated by the BOP agent Table 4.1 shows average distance from the counteroffers to the Pareto frontier as explained in Section 4.6. Obviously, the BOP and the ScalableBOP agents outperform all other strategies by generating offers much closer to the Pareto frontiers. There is some decay in the closeness

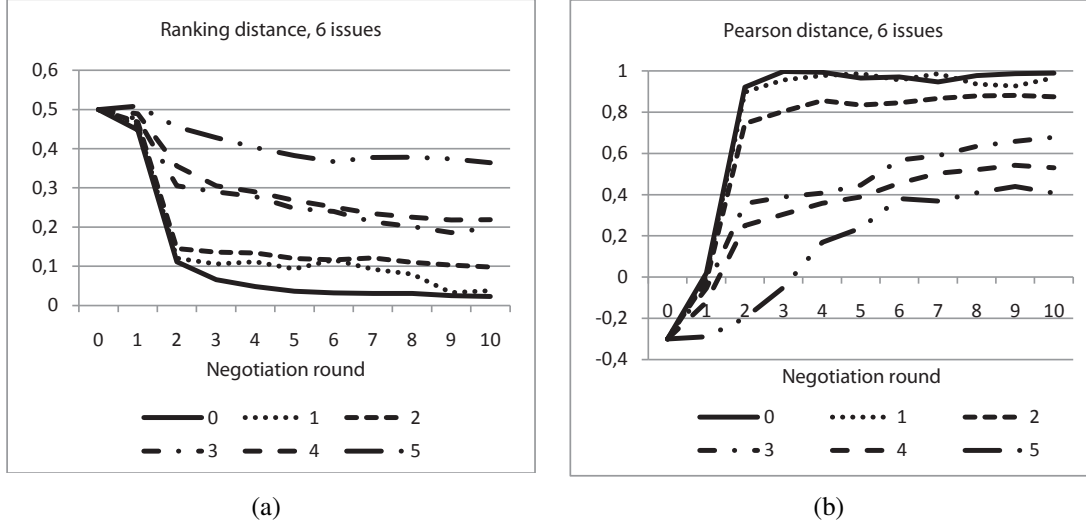


Figure 4.11: Learning quality measures for 6 issues domain with various numbers of unpredictable issues.

Number of issues	Strategy					
	ABMP	Trade-Off	TDT	ZI	BOP	ScalableBOP
4	0.32	0.15	0.35	0.45	0.03	0.06
6	0.29	0.18	0.33	0.52	0.04	0.08
8	0.36	0.24	0.34	0.54	n/a	0.11
10	0.38	0.26	0.34	0.53	n/a	0.13

Table 4.1: Average distance from the counteroffers to the Pareto frontier

of the counteroffers to the Pareto frontier in the large domains due to the decrease of the quality of learning as explained earlier.

To illustrate the impact of the BOP learning algorithm on the efficiency of the offers generated with its help two domains described in the negotiation literature were used. In the first domain, the setting is that of an employee and an employer who negotiate about a job assignment and related issues such as salary [20]. An interesting aspect of this domain is that both parties have the same preferences with regards to one of the issues. Figure 4.14 shows some results of the experiments with the Bayesian agent using BOP learning algorithm and other negotiation agents, including the resulting negotiation traces as well as the Pareto efficient frontier. The agreements reached are also marked explicitly. Table 4.2 shows average distance from the counteroffers to the Pareto frontier for all negotiation traces.

In this domain, the BOP 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 BOP agent needs little time to learn the issue evaluation functions and consecutively improves the weight esti-

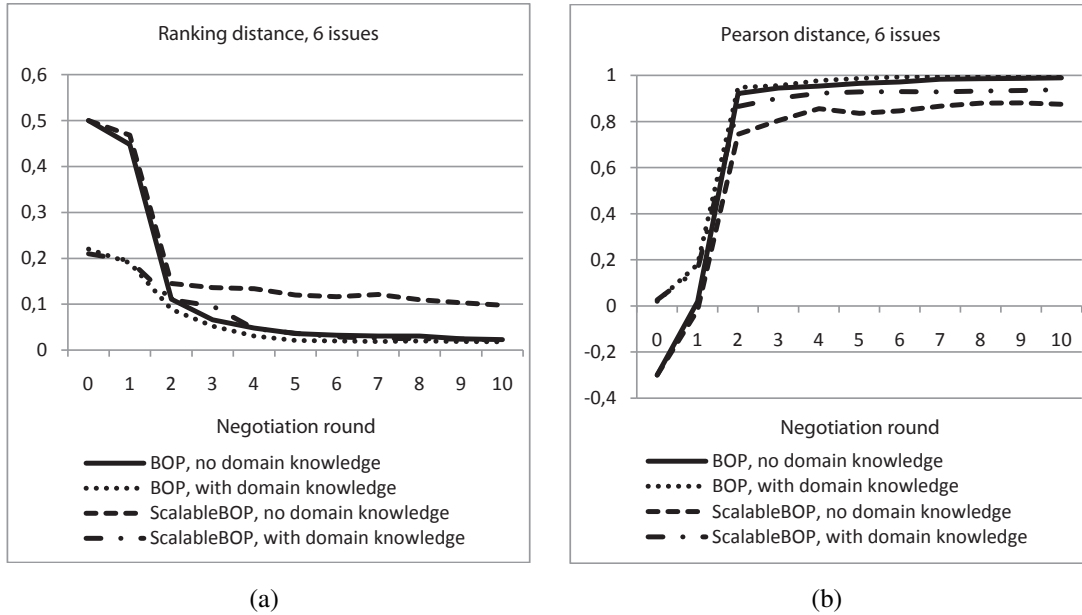


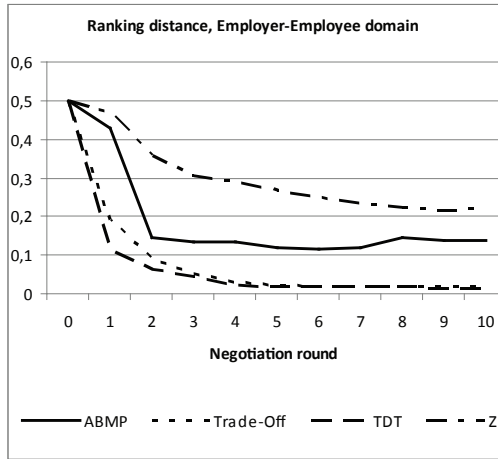
Figure 4.12: Learning quality measures for 6 issues domain with and without domain knowledge

Domain	Strategy							
	ABMP	TradeOff	TDT	ZI	Without dom.know.		With dom.know.	
					BOP	ScIBOP	BOP	ScIBOP
Employment	0.31	0.16	0.32	0.43	0.03	0.07	0.02	0.05
AMPOvsCity	0.35	0.19	0.34	0.51	n/a	0.11	n/a	0.08

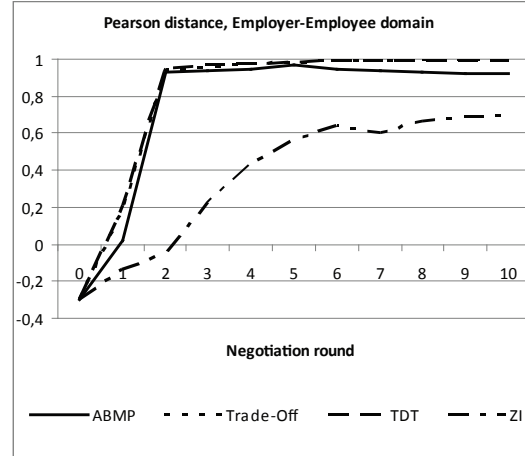
Table 4.2: Average distance from the counteroffers to the Pareto frontier for the Employer-Employee and AMPO vs Ciuty domains

mations. As a result the counteroffers generated by the BOP agents are close to the Pareto frontier. The Trade-off strategy, which uses domain knowledge but simply assumes that issue priorities are uniformly distributed, makes a number of unfortunate steps in this domain due to the fact that the parties consider different issues to be important. The counteroffers of the Trade-Off strategy have larger distance to the Pareto frontier than the BOP's counteroffers. Finally, the ABMP strategy is clearly outperformed by the strategy using Bayesian learning (see Figure 4.14 and Table 4.2 and almost uniformly concedes on all issues without considering the opponent's weights. ABMP lacks the capability of trading-in less important issues for more important ones. Since the Trade-off strategy is influenced by the efficiency of the opponent's strategy, it performs less efficient against the ABMP strategy. Note that only the Bayesian agents were able to reach an agreement close to the Pareto efficient frontier.

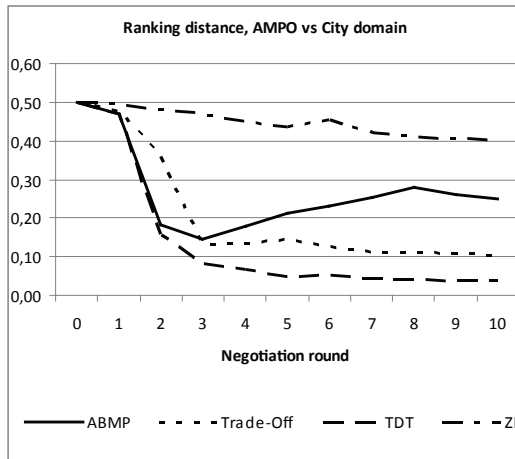
For the ScalableBOP a larger domain is used: the AMPO vs. City domain of [19], which consists of 10 issues, 5 values in average each (total of 7,128,000 possible outcomes).



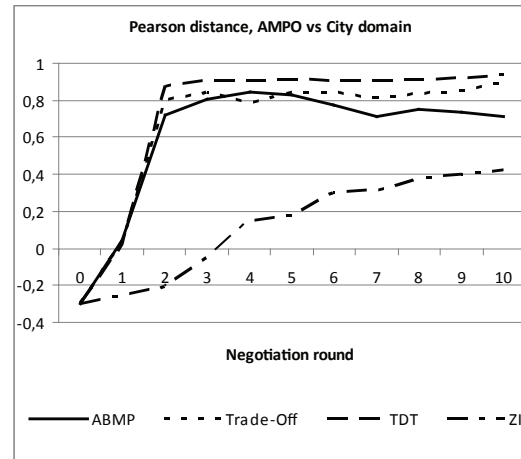
(a)



(b)



(c)



(d)

Figure 4.13: Learning quality measures for the Employer-Employee domain (a), (b) and the AMPO vs City (c), (d)

The results on this domain presented in Figure 4.15 show, as is only to be expected, that it becomes harder to stay close to the Pareto efficient frontier. The Bayesian learning agents stay a bit closer to the Pareto frontier than the agent based on the Trade-Off strategy. The ABMP strategy shows similar behaviour as on the earlier negotiation domains, and is outperformed by the other strategies. Also, note that the agreement reached by the Bayesian agents has a higher utility than that reached by the other strategies and that both the Bayesian agent without domain knowledge as well as the Trade-off agent make quite big unfortunate steps.

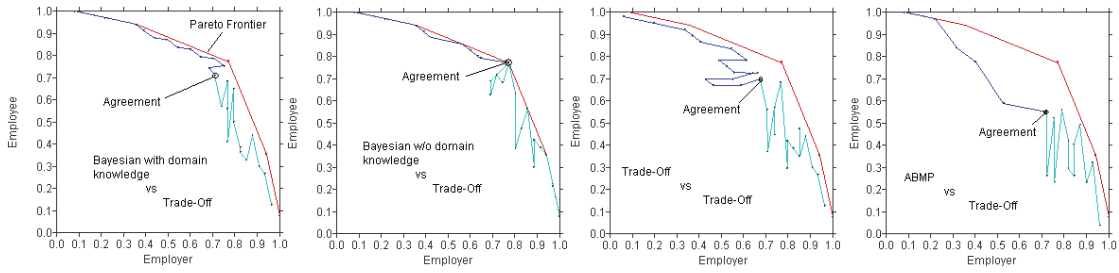


Figure 4.14: Negotiation dynamics for the Employee-Employer domain

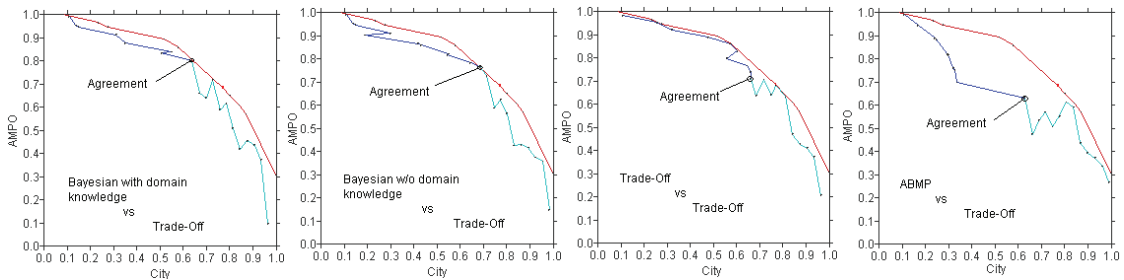


Figure 4.15: Negotiation dynamics for the AMPO vs. City domain

4.8 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.

To test the learning approach a method for the analysis of the learning quality of opponent preference profiles in automated negotiation was proposed. The method consists of three components: (i) It uses distance measures between the actual preference profile of the opponent and the learned preference profile to assess the quality of the learned model; (ii) it proposes criteria for the systematic classification of negotiation domains and preferences profiles to assess the impact of a variety of domains on the quality of the learned model; and (iii) it proposes some criteria to select a set of negotiation strategies.

The results of applying the proposed method to the BOP algorithm show that it is capable of learning the most important aspects of an opponent's profile in a single negotiation session, i.e., ranking of the weights of the issues and approximations of the evaluation functions per issue. The quality of learning in the proposed learning algorithm does not depend on the opposition of the preference profiles of negotiation parties. Therefore, the opponent profiles can range from cooperative to competitive.

The algorithm uses hypotheses about evaluation functions of issues used by the opponent based on their predictability. The quality of the learned profile improves along with the number of predictable issues. However, in negotiation domains with many unpredictable issues the BOP algorithm is still able to learn the opponent's preferences to some extent due to the approximative power of the proposed types of hypotheses.

The learning algorithm is robust in learning the profiles with respect to the opponent strategy. This holds for the e-negotiation strategies existing in the literature, as well as for a number of specific strategies designed to challenge the learning algorithm, such as Zero Intelligence.

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 is useful to increase the efficiency of learning a correct opponent model in the scalable learning algorithm proposed. One interesting line of future research is to test and initialize the learning algorithm for specific domains with an "average preference profile" derived from (large sets) of negotiator profiles for that domain. It is expected that performance of the algorithm on specific domains can be further enhanced. We are currently setting up an experiment to collect preference profiles for a negotiation domain and will test how our learning algorithm performs when it is initialized with such an aggregated profile.

The size of the hypotheses space in the standard algorithm is exponential in the number of issues. As such, the standard algorithm is not scalable with respect to the number of issues. Therefore, we developed and tested a simplified version of the algorithm, called ScalableBOP in which the number of hypotheses is linear. The difference is that the probability per hypothesis for one issue is updated independently of the probabilities for the other hypotheses. The speed of the learning and the final quality of the learned profile is less for ScalableBOP, but the effect is still highly significant.

The learning algorithm BOP deployed in BA can be integrated with any negotiation strategy that uses an opponent preference profile. The BA agent combines BOP with the Smart strategy of [5]. The results of using BA in a negotiation setting showed the effectiveness of using an opponent model in a negotiation strategy to improve the efficiency of the bidding process. We also discussed how to create QO-BOB that integrates BOP into the QO agent of [13].

In future work we will further investigate the expressiveness of the hypotheses space used in BOP to represent the opponent's preferences model and enable learning of more complex evaluation functions of an issue. Simply extending the hypotheses space by adding new types of evaluation functions is not a good solution because this would dramatically increase the size of the space. Various classes of functions that have been successfully used in approximation theories are good alternative candidates for learning the opponent's preferences.

An interesting direction for future work is to study the quality of learning of BOP on negotiation domains with interdependent issues. In such domains, the value of one issue

can influence the evaluation of another issue. Interdependencies between issues increase the complexity of the learning task. However, we believe that the BOP learning algorithm can handle some of the domains with interdependencies because in real-life cases a profile can be modeled by utility functions that are far from “wild” and they have a structure that can be captured by the hypotheses space of BOP. Furthermore, we would like to investigate how the hypotheses space can be extended to improve learning of the opponent’s preferences in negotiation domains with interdependent issues.

In future we plan to integrate BOP in our negotiation support system the Pocket Negotiator that offers bidding advice to human negotiators. A model of the opponent’s preferences learned by BOP can be used to improve efficiency of the human’s offers finding an alternative offer that would dominate the human user’s offer. In real-life negotiations a domain is not fixed during negotiation and new issues can be discovered by joint exploration of the negotiation space. Therefore, in future we plan to design an adaptable version of BOP that can handle a change in the number or structure of issues.

The BOP algorithm can be initialized with a priori knowledge about the most probable preferences in a specific negotiation domain. In some cases such knowledge can be learned from previously stored negotiations. Clustering techniques can be applied to the opponent’s model learned by BOP to find typical opponent profiles for a specific domain, and recurring themes over domains. The Pocket Negotiator can store such opponent profiles in a repository that can be used in future in case of a recurrent negotiation with the same opponent or with a different opponent in the same domain.

Bibliography

- [1] S. Buffett and B. Spencer. Learning opponents preferences in multi-object automated negotiation. In *In Seventh International Conference on Electronic Commerce (ICECS’05)*, pages 300–305. ACM, 2005.
- [2] R. Carbonneau, G.E. Kersten, and R. Vahidov. Predicting opponent’s moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2):1266–1273, February 2008.
- [3] R.M. Coehoorn and N.R. Jennings. Learning an opponent’s preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings of the 6th International Conference on E-Commerce*, pages 59–68, 2004.
- [4] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [5] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
- [6] Dhananjay K. Gode and Shyam Sunder. Allocative efficiency in markets with zero

- intelligence (zi) traders: Market as a partial substitute for individual rationality. *In Journal of Political Economy*, 100(1):119–137, 1993.
- [7] Vu Ha and Peter Haddawy. Similarity of personal preferences: Theoretical foundations and empirical analysis. *Artificial Intelligence*, 146(2):149–173, 2003.
 - [8] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Analysis of negotiation dynamics. In *Proceedings of 11th International Workshop CIA 2007 on Cooperative Information Agents, Delft, The Netherlands*, volume 4676 of *Lecture Notes in AI*. Springer-Verlag, 2007. Invited paper.
 - [9] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
 - [10] Catholijn M. Jonker and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. . *Autonomous Agents and Multi-Agent Systems Journal*, 15:221–252, 2001.
 - [11] Gregory E. Kersten and Sunil J. Noronha. Rational agents, contract curves, and inefficient compromises report. Interim Report IR-97-70, International Institute for Applied Systems Analysis, 1997.
 - [12] Raymond Y.K. Lau, Yuefeng Li, Dawei Song, and Ron Chi Wai Kwok. Knowledge discovery for adaptive negotiation agents in e-marketplaces. *Decision Support Systems*, 45(2):310–323, May 2008.
 - [13] Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence Journal*, 172(6-7):823–851, 2008.
 - [14] Raz Lin, Yinon Oshrat, and Sarit Kraus. Investigating the benefits of automated negotiations in enhancing people’s negotiation skills. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, May 2009.
 - [15] N. Matos, C. Sierra, and N. R. Jennings. Determining successful negotiation strategies: An evolutionary approach. In *Proceedings of the Third International Conference on Multi Agent Systems (ICMAS’98)*, pages 182–189, 1998.
 - [16] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
 - [17] T. Niimura, K. Ozawa, and O. Hirano. Fuzzy set-based decision support system for transactions of electricity in a deregulated environment. In *IFSA World Congress*, pages 87–92, 2001.
 - [18] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
 - [19] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.

- [20] L.L. Thompson. *The Heart and Mind of the Negotiator*. Pearson Prentice Hall, 2004.
- [21] D. Zeng and K. Sycara. Benefits of learning in negotiation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997.
- [22] D. Zeng and K. Sycara. Bayesian learning in negotiation. , *International Journal of Human Computer Systems*, 48:125–141, 1998.

Chapter 5

The Benefits of Opponent Models in Negotiation

Abstract. Information about the opponent is essential to improve automated negotiation strategies for bilateral multi-issue negotiation. In this paper we propose a negotiation strategy that exploits a technique to learn a model of opponent preferences in a single negotiation session. An opponent model may be used to achieve at least two important goals in negotiation. First, it can be used to recognize, avoid and respond appropriately to exploitation, which differentiates the strategy proposed from commonly used concession-based strategies. Second, it can be used to increase the efficiency of a negotiated agreement by searching for Pareto-optimal bids. A negotiation strategy should be efficient, transparent, maximize the chance of an agreement and should avoid exploitation. We argue that the proposed strategy satisfies these criteria and analyze its performance experimentally.

5.1 Introduction

In bilateral negotiation, two parties aim at reaching a joint agreement, by exchanging various offers using e.g. an alternating offers protocol. Two basic, constitutive facts about negotiation define the basic dilemma each negotiator has to face: (1) each party aims to satisfy its own interests as best as possible, but (2) in order to reach an agreement one has to take one's opponent's preferences into account as well.

In the literature on automated negotiation, typically, concession-based strategies have been proposed. An agent that uses a concession-based strategy selects as the next offer it will make an offer that has a decreased utility compared with the last offer made. The utility that is being decreased is the utility from the agent's own perspective without any guarantee that such a decrease will also increase the utility from the other party's perspective. A well-known example of such a strategy is the time-dependent strategy which

decreases utility simply as a function of time [3]. Although motivated by fact (2) above, such strategies do not explicitly take the opponent's preferences into account, and, as a result, will most likely be inefficient in complex negotiation domains. Moreover, time-dependent strategies can be exploited by the other negotiating party and as such do not adequately take fact (1) above into account.

The solution to these problems is to explicitly take the preferences of an opponent into account. The benefits of doing so are that it enables a search through the outcome space for outcomes that are mutually beneficial and that it allows classifying and recognizing the type of move an opponent has made. In order to do so, two key questions need to be addressed: How can an agent obtain information about opponent preferences? And: How can an agent exploit information about opponent preferences effectively?

In this paper we consider single session negotiations, i.e., negotiators cannot learn from repeated sessions with the same opponent. As negotiators typically are not willing to reveal their preferences to avoid exploitation, information about opponent preferences needs to be obtained from the behaviour of that opponent. The first question is addressed by means of opponent modelling techniques, several of which have been proposed, see e.g. [2, 11]. We use a technique based on Bayesian learning here that is able to effectively learn opponent preferences during a single negotiation session [6]. This paper shows how opponent preferences can be strategically exploited in negotiation. It is organized as follows. Section 5.2 discusses related work. In Section 5.3 a design of a negotiation strategy that explicitly uses opponent preferences is introduced. The theme of Section 5.4 is the algorithm of the proposed negotiation strategy. Its effectiveness is validated in Section 5.5 by way of experimental results. Section 5.6 concludes the paper.

5.2 Related Work

In this Section we first discuss related work on negotiation strategies, and then we briefly discuss related work on learning and introduce the technique we used in our experiments.

The literature on negotiation strategies is extensive and we only discuss some examples to illustrate the variety of ideas that have been proposed to design such strategies. In [3] a range of decision functions that may be used to define a negotiation strategy are discussed, focussing on different aspects that may be relevant in a negotiation such as time and the behaviour of an opponent. As no single tactic or decision function seems to be "right" in arbitrary negotiation settings, an approach proposed in [9] aims at combining different types of such negotiation tactics from [3] in a single strategy. An evolutionary algorithm is used to compute a next offer that adjusts the weights associated with each of the individual tactics. This approach is not suitable for the one-session closed negotiation situation we are focussing on. To begin with it requires a substantial number of negotiations to learn appropriate weights associated with the tactics. Moreover, the preference profiles of both parties must be made public in order to calculate the fitness during the learning phase. As

a result, the weights learned to combine the strategies only yield efficient negotiations in negotiation setups that are one-session and closed.

The negotiation strategy of [4] can be used in one-session closed negotiation setting. It postpones concessions by using domain knowledge to offer bids that increase the utility (and thus acceptability) for the opponent without decreasing the utility associated with the offer for the agent making the offer. Such offers increase the chances of a proposed trade-off that is good for both parties. Nevertheless, this strategy concedes even if the opponent does not.

The concession-based negotiation strategy of [8] determines the size of its next concession mainly on the basis of the utility gap between the last bids of the agent and the opponent. The next bid configuration, however, is based purely on the agent's own preference profile and thus reaches no win-win outcomes. Its time-dependent nature means that it can be exploited by the opponent.

To tackle the problem of exploitation, in [3], a number of variants of Tit-for-Tat tactics are discussed that belong to the family of so-called behaviour-dependent tactics. These tactics, however, do not use an opponent model and only vary utility of the agent's own perspective consistently. These tactics thus are blind to the preferences of an opponent. The behaviour generated by such a tactic therefore is not transparent and may be hard to understand from the opponent's point of view. With Axelrod [1984], we consider transparency an important feature of any strategy, which has motivated the design of the strategy introduced here. Transparency may be achieved by using available knowledge about the preference profile of the opponent, as explained in Section 5.3.

Various approaches to learning in a negotiation context have been based on forms of Bayesian learning, e.g. [6, 11]. For the negotiation situations we are focusing on, we need a technique that is able to learn the opponent profile during one session, such as the Bayesian learning technique introduced in [6], which we chose as a building block in this paper.

5.3 Negotiation Strategy Design

The preferences of an opponent can be used in at least two ways. First, it can be used to propose efficient Pareto-optimal offers. Finding such offers requires that the Pareto frontier can be approximated which is only feasible if a reasonable model of the opponent's preferences is available. Second, it can be used to recognize and avoid exploitation. The strategy we propose is inspired by a classification of negotiation moves as described in [5] and the Tit-for-Tat tactic, discussed in [1] and - in a negotiation context - in [3]. As learning techniques will not provide perfect models of an opponent's preferences a strategy should be robust with respect to such imperfections. We return to this last point in the Section 5.5. The design of the negotiation strategy proposed in this paper is based on a number of observations and criteria that we want the strategy to satisfy. The main criteria

are that the strategy should be efficient, transparent, maximize the chance of an agreement and should avoid exploitation.

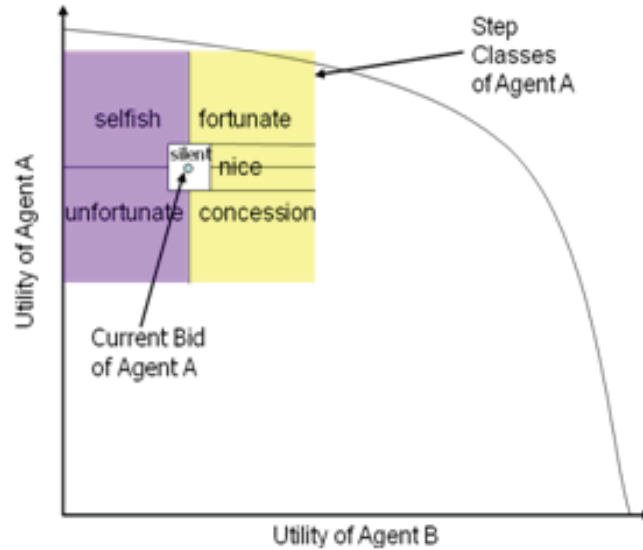


Figure 5.1: Classification of negotiation moves

The first observation relevant to the design of our strategy is that the availability of information about the preferences of an opponent enables an agent to classify the moves its opponent makes. Here, we use a classification of moves proposed in [5] and illustrated in Figure 5.1. The move classification is presented from the perspective of agent A.

Given that agent A's last offer is marked by the arrow "Current Bid of Agent A", the agent has a number of choices for making a next negotiation move. A silent move does not change utility of either party significantly. A concession move decreases own utility but increases the utility of the opponent. A fortunate move increases utility for both parties whereas an unfortunate move does the opposite. Note that a fortunate move can only be made if the current bid is not already on the Pareto frontier. A selfish move increases own utility but decreases the opponent's utility. Finally, a nice move increases the opponent's utility but does not change the agent's own utility.

Based on this classification a simple suggestion would be to "mirror" each move of an opponent by making a similar move, which would implement a Tit-for-Tat-like tactic. The basic idea of a Tit-for-Tat strategy in a multi-issue negotiation context would be to respond to an opponent move with a symmetrical one. That is, "match" the move as depicted in Figure 5.2 by mirroring it in the diagonal axis.

First note that each type of move would indeed result in a response move in the same class. In particular, responding to a concession move of the opponent with a concession move itself arguably is one of the most reasonable responses one can make. All rational

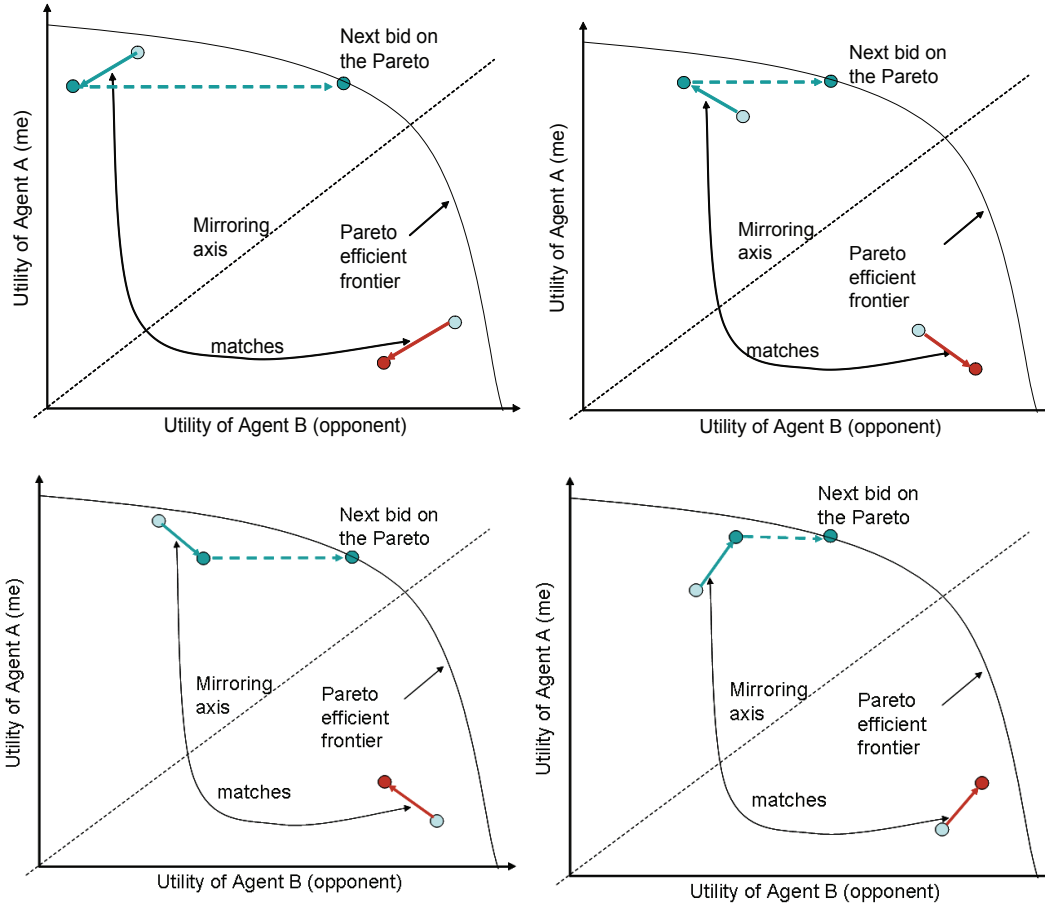


Figure 5.2: Classification of negotiation moves

negotiation strategies will attempt to make concession moves at some point during a negotiation. Moreover, the “mirroring” strategy would avoid exploitation as a selfish move of the opponent would result in a selfish response move. Such a response would be a signal to the opponent: “I am prepared to make a concession towards you only if I get something in return. If you pull back I’ll do the same”.

A mirroring strategy would, however, be too simplistic for several reasons. A mirroring strategy is not rational in the case of an unfortunate move, as there is no reason to decrease the agent’s own utility without increasing the chance of acceptance of the proposed bid by the opponent. Furthermore, observe (compare Figure 5.2) that unfortunate moves move away from the Pareto-optimal frontier, and thus would not satisfy our efficiency criteria.

In order to remove these deficiencies, we propose to first mirror the move of the opponent and thereafter make an additional move towards the Pareto frontier, i.e. a move towards the approximated Pareto frontier that is computed using the learned opponent model and the agent’s own preference profile. There are multiple ways to do this and the choice is not straightforward. What is clear is that the move towards the Pareto frontier should not further decrease the agent’s own utility as this would invite exploitation tactics. Further-

more, it also does not seem rational to further decrease the opponent's utility as this would result in selfish moves to arbitrary moves of the opponent.

The final observation that motivated our choice is that increasing the agent's own utility by moving towards the Pareto frontier actually minimizes the chance of reaching an agreement when this strategy would be used by both parties, which would violate one of our design criteria for a negotiation strategy. To explain this, consider two agents that would mirror an opponent's move and then, seen from the perspective of Agent A in Figure 5.2, would move straight up towards the Pareto frontier (Agent B would move right) which would only increase own utility. The other agent in this case would consider such a move a selfish move and respond similarly, thereby minimizing the chance of reaching an agreement. Of course, this line of reasoning depends on the quality of the opponent model but presents a real problem. To resolve it, the strategy we propose only increases the opponent's utility when moving towards the Pareto frontier in order to maximize the chance of an agreement. The resulting strategy consists of two steps: first mirror the move of the opponent and then add a nice move to propose an efficient offer (i.e., search for a bid on the approximated Pareto frontier that is on the same iso-curve as the bid obtained by mirroring, see Figure 5.2). This strategy we call the Mirroring Strategy (MS). To gain a better understanding of MS, it is instructive to discuss some of the response moves MS generates. Figure 5.2 shows examples of responses to an unfortunate, selfish concession and fortunate move. The response to an unfortunate move is to mirror this move and add a nice move, which results in a concession move (see Figure 5.2a). This is a reasonable reply as such a move may be interpreted as an attempt (that failed) to make a concession move by the opponent (due to the lack of information about the preferences of its opponent). Such a move which is the result of misinformation should not be punished, we believe, but an attempt instead should be made to maintain progress towards an agreement.

The response to a selfish move either results in a fortunate move or in a selfish move. Figure 2b shows the case resulting in a fortunate move. It should be noted that a fortunate move is only possible if the previous move the agent made was inefficient. This means that in that case the opponent model must have misrepresented the actual preferences of the opponent. In such a case, where our previous move was based on misinformation, we believe it is reasonable to not punish the opponent with a selfish move and give the opponent the benefit of the doubt in such a case. If, however, the previous move would have been efficient, a selfish move most likely would be replied to with a selfish move (since there would be no room to make a nice move towards the Pareto frontier), and it is reasonable to send a clear signal to the opponent that such moves are unacceptable.

Finally, both a concession move as well as an unfortunate move of the opponent would be replied to with the same type of move (see Figure 5.2c and 5.2d). Moreover, if there is room for a nice move towards the Pareto frontier, in both cases the step would be bigger than that made by the opponent, increasing the utility of the opponent even more and thereby again increasing the chance of acceptance as early on in a negotiation as possible.

As discussed, a negotiation strategy should be efficient, transparent, maximize the chance

of an agreement and should avoid exploitation. It is clear that MS aims to be as efficient as possible, which only depends on the quality of the learning technique for modelling opponent preferences. Performance of the learning algorithm used in MS was studied in [7]. The study concludes that the learning algorithm can learn the most important aspects of the opponent preferences in a range of negotiation settings. MS does not aim at exploiting the weaknesses of an opponent strategy. Instead it aims for restoring efficiency whenever an opponent strategy is not able to do so and aims at a fair outcome (see also Section 5.5 below). MS is transparent as it proposes a simple response strategy by mirroring an opponent's move and then adding a nice step. The signals thus sent by negotiation moves are easy to interpret by an opponent. In particular, MS only punishes an opponent in reply to a selfish move and only does so when the model of opponent preferences matches the actual preferences of that opponent. As a result, MS not only avoids exploitation but also is a nice strategy. MS is nice even when an opponent makes unfortunate moves which are interpreted as "mistakes" on the opponent's part. The strategy moreover maximizes the chance of an agreement as early as possible, which is achieved by the move towards the Pareto frontier that always maximizes the utility of the opponent relative to a particular utility for the agent itself.

5.4 Matching Strategy Algorithm

Here we present the MS strategy in 6 algorithmic steps, including the steps needed for learning an opponent model. The algorithm is presented in Figure 5.3. As is usual, MS starts by proposing a bid that has maximal utility with respect to the agent's own preference profile (step 1). In step 2 a simple but reasonable acceptance strategy is used which is not particular to MS. A bid from an opponent is accepted when the utility of that bid is higher than that of the agent's own last bid or the utility of the bid it would propose next. Otherwise, the agent will propose a counter-offer. In step 3, the bid received from the opponent is used to update the opponent model with the function $\tilde{U}(\omega)$ of [4]. Steps 4, 5 and 6 define MS. Step 5 mirrors an opponent's move, after which step 6 determines a nice move towards the Pareto frontier (given the opponent model computed in step 3).

Note, that in the beginning of a negotiation the model of the opponent preferences is inaccurate because it has not been updated yet. However, this is not crucial at this stage of the negotiation given that both agents would start by offering a bid with the highest utility for themselves that is the most rational choice. The initial offer of the opponent is used to make the first update of the opponent model. The second move of the MS strategy can only be a concession or an unfortunate move because the initial offer cannot be improved for either player and, therefore, the other classes of moves are not possible. In this case the size of the first concession in the opponent's utility would be determined by the efficiency of the learning technique. Given the fact that MS always tries to maximize the opponent's utility it will try to make a concession move, thus signalling to the opponent its readiness to proceed in the same manner.

1	Start with a bid of maximal utility w.r.t. the agent's own preference profile: $\omega_0 = \arg \max_{\omega \in \Omega} U(\omega)$
2	Receive an opponent's bid ω_t^o. Accept the opponent's bid if its utility is at least high as the utility of own previous bid: $action = \begin{cases} accept, & U(\omega_t^o) \geq U(\omega_{t-1}) \\ go to step 3, & otherwise \end{cases}$
3	Update the opponent preference model given : $\omega_t^o : \tilde{U}_{t-1}(\omega) \xrightarrow{update} \tilde{U}_t(\omega)$
4	Classify the opponent's move ω'_{t-1} to ω'_t: <i>concession</i> : $\Delta U(\omega'_t) > 0$ and $\Delta \tilde{U}(\omega'_t) < 0$ <i>unfortunate</i>: $\Delta U(\omega'_t) < 0$ and $\Delta \tilde{U}(\omega'_t) < 0$ <i>selfish</i>: $\Delta U(\omega'_t) < 0$ and $\Delta \tilde{U}(\omega'_t) > 0$ <i>fortunate</i>: $\Delta U(\omega'_t) > 0$ and $\Delta \tilde{U}(\omega'_t) > 0$ where: $\Delta U(\omega'_t) = \Delta U(\omega'_t) - \Delta U(\omega'_{t-1})$ $\Delta \tilde{U}(\omega'_t) = \Delta \tilde{U}(\omega'_t) - \Delta \tilde{U}(\omega'_{t-1})$
5	Calculate coordinates of the matching bid ω_t', given the classification in the previous step: <i>fortunate, unfortunate</i>: $\Delta U(\omega_t^o) = \Delta U(\omega'_t); \quad \Delta \tilde{U}(\omega_t^o) = \Delta \tilde{U}(\omega'_t)$ <i>selfish, concession</i>: $\Delta U(\omega_t^o) = -\Delta U(\omega'_t); \quad \Delta \tilde{U}(\omega_t^o) = -\Delta \tilde{U}(\omega'_t)$ Then the coordinates of ω_t' are: $U(\omega'_t) = U(\omega'_{t-1}) + \Delta U(\omega'_t)$ $\tilde{U}(\omega'_t) = \tilde{U}(\omega'_{t-1}) + \Delta \tilde{U}(\omega'_t)$ If $U(\omega'_t) < \text{reservation value}$ then stop negotiation
6	Find a bid ω_t that corresponds to the ω'_t and belongs to the approximated Pareto frontier and belongs to the same iso-curve on the $U(\omega'_t)$: $\omega_t = \arg \max_{\omega \in \Omega, U(\omega) = U(\omega'_t)} \tilde{U}(\omega)$
7	Send bid ω_t to the opponent.
8	Go to step 2.

Figure 5.3: Matching Strategy algorithm

Domain	Utility spaces		Domain size	Number of predictable
	Ranking	Pearson		
AMPO vs. City	0.662	-0.482	7,128,000	3 (10)
Party	0.540	-0.126	3,125	0 (5)
SON	0.669	-0.453	810,000	4 (4)
2nd hand car	0.635	-0.387	18,750	1 (5)
Employment contract	0.698	-0.584	3,125	5 (5)

Table 5.1: Summary of the negotiation domains

The MS strategy is developed to avoid exploitation by the state-of-the-art rational strategies and tries to match the opponent’s moves in a transparent way as it is defined by the design criteria. The MS strategy is experimentally tested in a tournament setting against such strategies (see Section 5.5). To prevent exploitation by irrational strategies we use a reservation value to limit the concessions made by the MS strategy. The reservation value is defined in the user’s preference profile and represents a utility value below which all bids are unacceptable for the agent.

5.5 Experimental Analysis

We test the efficiency of the MS strategy in an experimental setup, in which the MS strategy negotiates against automated negotiation strategies available in the literature and against human negotiators. Furthermore, this Section shows that the MS strategy constructed results in a fair agreement for both parties.

The AMPO vs City domain [10] is the largest domain in our test. The Party domain developed by us is small with rather cooperative preference profiles. Humans tend to perform well on this domain. The Service-Oriented Negotiation (SON) domain was taken from [4]. The Employment contract negotiation domain was taken from [9]. The 2nd hand car domain was taken from [8].

Besides MS four other strategies were used. The Zero Intelligence (ZI) strategy randomly proposes bids above its break-off point, which was set to 0.6 in the tournament. It is difficult for the ZI strategy to achieve a better agreement than its break-off point and any effective negotiation strategy should be expected to at least outperform it. We use the ZI strategy as a baseline. It also provides a good test case for any learning technique. Details about the ABMP strategy can be found in Section 5.2 and in [8], for the Trade-off strategy in [4] and Section 5.2. The Bayesian Smart strategy is similar to the Trade-off strategy but uses the Bayesian learning algorithm from [6] to model opponent preferences. As the same learning algorithm has been used by MS, the Bayesian Smart strategy can be used to compare performance of MS with that of the Bayesian Smart. To analyze the robustness of MS in negotiations against humans an experiment was setup in which 42 subjects first negotiated face-to-face and then negotiated against MS that used the same profile as the

human opponent in the first session. The Party domain was used for the experiment. The human subjects were able to familiarize themselves with a negotiation environment used in the experiment and practice on other domains.

For every negotiation domain and preference profile, the utilities of agreements obtained by a strategy against all other strategies in the tournament were averaged. The ZI strategy was used as a baseline. Table 5.4 reports the percentage increase compared to the average utility of this strategy.

Negotiation Domain	Negotiation Strategy			
	ABMP	Trade-Off	Bayesian Smart	NMS
Car	16%	12%	13%	14%
Party domain	13%	9%	13%	14%
Service Oriented	14%	17%	25%	38%
AMPO vs City	10%	13%	14%	20%
Employment contr.	11%	40%	44%	47%

Figure 5.4: Utility increase relative to the ZI strategy

MS shows improved performance compared with the benchmark Bayesian Smart strategy on all domains. The main reason is that MS is more robust since it matches the moves of its opponent and does not concede more than its opponent. The results show that on all domains MS outperforms the other strategies, except for the 2nd hand car domain where ABMP performs best. The differences on this and the Party domain are not big for all strategies.

The most significant improvement compared to ZI is achieved in the Employment contract domain. This domain is relatively small and issues are predictable. Learning an opponent model is relatively easy, and important in this domain as it contains compatible issues (i.e., both agents have similar preferences with regard to such an issue).

The results on the SON and AMPO vs City domain are comparable to that of the Employment domain. It is more difficult to reach efficient agreements in the SON domain as this domain is bigger and the variation of issue importance is much bigger. The performance of MS on the AMPO vs City domain is not as good mainly due to the decreasing performance of the learning technique in domains of high dimensionality. The improvement over the benchmark Bayesian Smart strategy is still significant in both these domains which shows that MS is a robust strategy even when the model of the opponent's preferences is not very good. Improvement is caused by the fact that MS tries to match the opponent's moves, which, at least with respect to own utility it is always able to do. Therefore, even if the quality of the learned model is low as it is e.g. in the AMPO vs City domain, MS unlike the Bayesian Smart strategy will concede only if the opponent does so too.

To analyze the robustness of MS more precisely we consider the results on the SON domain as shown in Table 5.5. The quality of learning is high in this domain and, therefore,

is a good choice to test the robustness of MS against various strategies. Table 5.5 lists average utility values of the agreements reached for each party in the tournament on the SON domain. We have used the standard deviation of these utilities as a measure of the robustness of MS. The average utility value of agreements is high and the deviation of the utility of agreements is lower for MS than other strategies, which confirms that MS is more robust and more difficult to exploit than these strategies.

The technique used by the Trade-Off strategy to match the opponent's preferences strongly depends on the efficiency of the strategy used by the opponent, see [5]. E.g., the Trade-Off strategy is not able to find Pareto efficient offers in settings where it negotiates against less efficient strategies such as the ABMP and the ZI strategy. As a result, the utility of outcome reached by the Trade-Off strategy is in average higher than that of the ZI and ABMP strategy, but its deviation is relatively high (see Table 5.5). The learning technique used in the MS strategy, on the other hand, does not depend on the efficiency of the opponent's strategy (see [7]) and, therefore, is able to achieve better results in negotiations against the ZI and the ABMP strategy.

Role	Utility Statistics	ZI	ABMP	Trade-Off	Bayesian Smart	NMS
Agent A	Average	0.574	0.657	0.726	0.748	0.769
	Std. dev.	0.056	0.079	0.103	0.023	0.020
Agent B	Average	0.519	0.657	0.652	0.800	0.805
	Std. dev.	0.030	0.066	0.134	0.044	0.043

Figure 5.5: Utility of agreement in the SON domain

Furthermore, we report on the performance of MS in an experiment with humans. The performance of MS in an experiment with human subjects also shows it is a robust strategy. Human subjects were not able to exploit NMS on the Party domain. Overall human performance was very good and close to the Pareto frontier due to simplicity the domain. Even so the humans had an advantage of training in the first face-to-face negotiation session, still MS managed to improve average utility with 5%, whereas in 30% of the experiments the increase in utility was larger than 5%.

Finally, as MS tries to match the moves of an opponent, it is reasonable to assume that MS typically tends to result in a fair outcome. This hypothesis is confirmed by Table 5.6, which shows that the agreements reached by MS are, on average, closer to the Nash and Kalai-Smorodinsky solutions on all domains. The results also show that MS prefers the Kalai-Smorodinsky over the Nash solution.

Negotiation Domain	Negotiation Strategy				
	ZI	ABMP	Trade-Off	Bayesian Smart	NMS
Distance to Kalai-Smorodinsky solution					
Car	0.20	0.13	0.12	0.12	0.11
Party domain	0.23	0.15	0.13	0.12	0.12
Service Oriented	0.25	0.23	0.16	0.14	0.11
AMPO vs City	0.20	0.15	0.13	0.13	0.12
Employment contr.	0.26	0.26	0.14	0.14	0.14
Distance to Nash Point					
Car	0.19	0.15	0.14	0.14	0.13
Party domain	0.20	0.19	0.15	0.13	0.13
Service Oriented	0.26	0.26	0.19	0.17	0.16
AMPO vs City	0.23	0.24	0.20	0.18	0.17
Employment contr.	0.26	0.26	0.14	0.14	0.14

Figure 5.6: Utility of agreement in the SON domain

5.6 Conclusions

The Nice Mirroring Strategy introduced here shows that two important goals in closed multi-issue negotiations can be achieved when a reasonable estimate of the preferences of an opponent is available. Using a learning technique to obtain such an opponent model, it is possible to increase the efficiency of the negotiated agreement and to avoid exploitation by the other party.

The design of MS has been based on a classification of negotiation moves developed for the analysis of negotiation strategies, see [5]. MS satisfies several design criteria we believe are important for any negotiation strategy. A negotiation strategy should be efficient, transparent, maximize the chance of an agreement and should avoid exploitation. MS has been shown to be efficient and fair as it is biased towards the Kalai-Smorodinsky solution. MS is transparent as it proposes a simple response strategy by mirroring an opponent's move and then adding a nice step, i.e. a move over the utility iso-curve towards the Pareto frontier. In fact, MS is a "nice" strategy as it will only punish an opponent in reply to a selfish move and only does so when the model of opponent preferences matches the actual preferences of that opponent. The strategy moreover maximizes the chance of an agreement as early as possible, which is achieved by the move towards the Pareto frontier, that always maximizes the utility of the opponent relative to a particular utility for the agent itself. The effectiveness of MS has been validated experimentally in a tournament setup, using domains of different characteristics and a number of different

negotiation strategies. The results show that MS is able to realize significant increases in utility. In future work, we plan to investigate the exploitability of strategies and in particular MS. An important related theme for future work concerns acceptance criteria used by negotiation strategies.

Bibliography

- [1] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.
- [2] R.M. Coehoorn and N.R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings of the 6th International Conference on E-Commerce*, pages 59–68, 2004.
- [3] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [4] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
- [5] Koen Hindriks, Catholijn Jonker, and Dmytro Tykhonov. Analysis of negotiation dynamics. In *Proceedings of 11th International Workshop CIA 2007 on Cooperative Information Agents, Delft, The Netherlands*, volume 4676 of *Lecture Notes in AI*. Springer-Verlag, 2007. Invited paper.
- [6] Koen Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the AAMAS 2008*, 2008.
- [7] Koen Hindriks and Dmytro Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. In *Proceedings of 12th Int. Workshop on Cooperative Information Agents*. Springer-Verlag, 2008.
- [8] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
- [9] J. Nadler, L. Thompson, and L. van Boven. Learning negotiation skills: Four models of knowledge creation and transfer. *Journal of Management Science*, 49(4):529–540, 2003.
- [10] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.
- [11] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human Computer Systems*, 48:125–141, 1998.

Chapter 6

Qualitative One-to-Many Multi-Issue Negotiation: Approximating the QVA

Abstract. When there is one buyer interested in obtaining a service from one of a set of sellers, multi-attribute or multi-issue auctions can ensure an allocation that is efficient. Even when there is no transferable utility (e.g., money), a recent qualitative version of the Vickrey auction may be used, the QVA, to obtain a Pareto-efficient outcome where the best seller wins. However, auctions generally require that the preferences of at least one party participating in the auction are publicly known, while often making this information public is costly, undesirable, or even impossible. It would therefore be useful to have a method that does not impose such a requirement, but is still able to approximate the outcome of such an auction.

The main question addressed here is whether the Pareto-efficient best-seller outcome in multi-issue settings without transferable utility (such as determined by the QVA) can be reasonably approximated by multi-bilateral closed negotiation between a buyer and multiple sellers. In these closed negotiations parties do not reveal their preferences explicitly, but make alternating offers. The main idea is to have multiple rounds of such negotiations. We study three different variants of such a protocol: one that restricts the set of allowed offers for both the buyer and the seller, one where the winning offer is announced after every round, and one where the sellers are only told whether they have won or not after every round. It is shown experimentally that this protocol enables agents that can learn preferences to obtain agreements that approximate the Pareto-efficient best-seller outcome as defined by the auction mechanism. We also show that the strategy that exploits such a learning capability in negotiation is robust against and dominates a Zero Intelligence strategy. It thus follows that the requirement to publicly announce preferences can be removed when negotiating parties are equipped with the proper learning capabilities and negotiate using the proposed multi-round multi-bilateral negotiation protocol.

6.1 Introduction

In a procurement setting in which a buyer faces several sellers an auction may provide an effective mechanism to reach an agreement. Auctions may also be used when the outcome that needs to be reached is complex and consists of multiple issues that need to be settled, such as when a Request For Quote (RFQ) is issued by a corporation or government organization [19, 20]. A bid in such a reverse auction may involve, for example, the desired quality of service, the quantity demanded, the terms and time of delivery, and so forth. Such a setting with one buyer and multiple sellers (i.e., a reverse auction) is used throughout this paper, but all results directly transfer to a forward auction with one seller and multiple buyers as well.

The various types of auctions have desirable theoretical properties such as yielding an efficient outcome and being strategy-proof. However, some of these mechanisms impose requirements which are not easy to meet in practice. One of these requirements generally associated with (reverse) auctions is that the preferences of the buyer have to be known by all bidders. This requirement is often not realistic in practice. First of all, the explicit elicitation of a buyer's value function may be difficult [2]. Even modeling such preferences is a very complex problem, and very relevant in the context of auctions [19]. The buyer may not know the complete domain of possible outcomes as sellers may come up with new options during the process, and it usually is very hard to specify preferences completely over a complex and possibly infinite set of outcomes. This is particularly true for auctions that are used to settle multiple issues, e.g., related to an RFQ. Finally, the buyer may not want to publicly reveal his preferences to the extent required by multi-issue auctions. It may be disadvantageous to do so given that it is not unlikely that future encounters with similar parties will take place.

In negotiations, on the other hand, the preferences of one party are only partially revealed to the other in the course of the process. However, in settings where one buyer may make a choice among a set of sellers, the competitive aspect is not explicitly taken into account in negotiation protocols. In recent work on multiple (parallel) bilateral negotiations [14, 17, 13], this is partly taken care of by informing all other sellers when a provisional agreement is reached in one of the negotiation sessions. However, when other sellers improve upon this outcome, the seller who has reached the first agreement does not get any opportunity anymore to improve upon this. Such negotiation protocols consequently may often lead to inefficient agreements.

It thus becomes interesting to look for alternative methods that may be used that guarantee outcomes that approximate the efficient outcome of an auction mechanism. The problem we study in this paper is whether alternative mechanisms based on multiple bilateral (also called multi-bilateral) negotiations can be used to reduce the preference information that needs to be made public but that also retain some of the desired theoretical properties such as efficiency of agreements as guaranteed by the auction mechanism. Studying mechanisms based on multilateral negotiations is interesting in its own right [21], but also because their relationship to various auction formats has implications for institu-

tional design. Studying the factors that relate and differentiate auctions from negotiation mechanisms may lead to a more informed selection of a transaction mechanism.

In this paper, we consider a particularly interesting instance of this more general one-to-many multi-issue allocation problem where there is not necessarily the possibility to transfer utility (such as when the government announces a fixed budget in a RFQ).¹ Without transferable utility, most of the known auction mechanisms cannot be used. However, one particular auction mechanism, called the *Qualitative Vickrey Auction* (QVA) [8], can obtain a Pareto-efficient outcome where the best seller wins. However, the QVA requires the buyer to publicly announce its preferences.

We study various multi-bilateral negotiation mechanisms. The main idea is that the (efficient) outcome of the QVA may be approximated by a negotiation protocol that consists of multiple negotiation rounds in which sellers are provided an opportunity to outbid the winner of the previous round. We show experimentally that each of these mechanisms is able to approximate the efficient outcome as defined by the QVA. The main assumption that we need to make to obtain this result is that the negotiating agents are able to (privately) learn part of the preferences of their opponents during a negotiation session. Techniques to do so are available [9], making our proposal one that can be implemented given the current state of the art in negotiation. Additionally, experiments are performed that show that a negotiating agent that exploits learning outperforms a Zero Intelligence strategy [6].

The paper is organised as follows. In Section 6.2 we define the general setting of a buyer and multiple sellers that aim to reach an agreement settling multiple issues. This setting is generic in the sense that it covers arbitrary situations where one buyer wants to obtain an efficient multi-issue agreement with any one out of a set of available sellers. Section 6.3 introduces the QVA auction that may be used to reach such an agreement. In Section 6.4 we then propose three variants of a multi-bilateral negotiation protocol as alternative mechanisms to the QVA. Each of these protocols is related to the QVA in the sense that it approximates the outcome defined by the QVA. The different protocols introduced moreover progressively require less information to be revealed publicly by the buyer. Section 6.5 presents experimental results to evaluate how well these protocols approximate the outcome defined by the QVA mechanism. The results validate our claim that the QVA may be replaced by a multi-bilateral negotiation protocol while still obtaining agreements that are similar. The tradeoff that has to be made concerns the amount of effort and time that needs to be invested in reaching an agreement. Finally, Section 6.6 discusses related work and Section 6.7 concludes with a discussion of the results obtained and outlines directions for future research.

¹This also means that 'pricing out' is not an option to elicitate preferences [19].

6.2 Definitions

The setting we consider in our work consists of a buyer that wants to procure a service or product from one out of a potentially large number of sellers. An agreement in this setting is an outcome that fixes the parameters of the service to be provided. Formally, the space of all possible outcomes is defined as all tuples $x = \langle x_1, \dots, x_m \rangle \in X$ over m issues in a domain $X = X_1 \times \dots \times X_m$. These issues define all aspects of the agreement, such as quality, start time, duration, guarantees, penalty, etc. Buyer and sellers are assumed to associate a *utility value* with each outcome and to have a *reservation value* that determines when an outcome does not improve the status quo for that party, i.e. the buyer or one of the sellers. In this paper we concentrate on a setting where we do not assume that one of these issues is a price. In other words, we relax the condition that there is a transferable utility (and we thus also relax the usual assumption that utility functions are quasi-linear in price).

We introduce the following notation. The buyer is denoted by 0 and sellers are denoted by $i \in \{1, \dots, n\}$. The *reservation value* of each party i is denoted by v_i and represents the minimal utility value that an agreement should have to be an acceptable outcome for that party. Outcomes with a utility below the reservation value are called *unacceptable*. Each party i also has a utility function $u_i : X \rightarrow \mathbb{R}$ which represents the utility that party associates with an outcome.

The goal is to find an agreement between the buyer and the sellers that is not only acceptable to both, but that is also Pareto efficient, i.e., there should not be another agreement with the same or higher utility for both players, and strictly higher for at least one of them. In addition to Pareto-efficiency of the final agreement between the buyer and the winning seller, we are interested in an agreement with the seller that can make the best offer to the buyer (that is still acceptable to the seller), often called allocative efficiency in the context of auctions. In this paper we call an outcome that meets both these efficiency conditions simply *efficient*.

An example is a buyer that is interested in buying a supercomputer. A range of potential suppliers is available that may provide a supercomputer. Apart from price (which is often limited by a given budget), supercomputers have many features (processing speed, memory, etc.) and requirements (regarding power supply, cooling, etc.) that need to be settled to obtain an agreement. Such an agreement thus is complex as many issues have to be agreed upon and finding an efficient outcome can be a complex process. In the next section an auction mechanism is summarized that has a dominant strategy equilibrium that yields a such an efficient outcome.

6.3 The Qualitative Vickrey Auction

The Qualitative Vickrey Auction (QVA) [7, 8] is particularly useful in a context where a single buyer tries to obtain a *complex agreement* with one out of many sellers that are interested in making such an agreement. This generalization of a Vickrey auction [22] is strategy-proof, and under most realistic settings (when money is involved, or the set of outcomes is discrete and linearly ordered by the buyer, or the preferences of the suppliers are equipeaked), it is efficient, i.e., it obtains a Pareto-efficient outcome that involves the seller that can make the best agreement (for the buyer) still acceptable to him.

Intuitively, this mechanism captures the negotiation power of the buyer. If there are many sellers, the buyer will end up with some very good offers, but if there is only one seller that has a sufficiently good offer, the agreement is not that good for the buyer. This interpretation can be given to most auction mechanisms. This mechanism, summarized below, has the special feature that it also works if none of the issues is about money.²

This auction mechanism can be thought of as consisting of two rounds. In the first round (1.1–1.3 below), the buyer publicly announces her preferences. Then potential sellers submit offers in response, and a winner is selected by the buyer. In the second round (2.1–2.2 below), the buyer determines the second-best offer (from her perspective again) she received from another seller, and announces this publicly. Finally, the winner is allowed to select any agreement that has at least the same utility to the buyer as the second-best offer (which can be determined by the winner since the preferences of the buyer are publicly announced). If the bids offered in the first round all are made public afterwards, anyone can check whether the buyer follows the protocol. The steps of this procedure can be found in Algorithm 1.

Algorithm 1 The qualitative Vickrey auction

1. Round 1: Winner selection
 - (a) The buyer announces her preferences.
 - (b) Every seller submits an offer.
 - (c) The buyer selects the winner according to her preferences.
 2. Round 2: Agreement selection
 - (a) The buyer announces the second-best offer she received.
 - (b) The winner may select any agreement that has at least the same utility for the buyer as the second-best offer.
-

The properties that make this mechanism interesting are not only Pareto efficiency, and that the seller wins that can make the best offer, but also that it is a dominant strategy for a seller to bid an offer that is just acceptable to itself and ranks highest in the buyer's preferences. In the problem domain defined in the previous section, this dominant strategy comes down to proposing an offer with exactly the same utility as its reservation value.

²If none of the issues is about money, a reverse auction is not different from a standard auction.

Formally, the winner in a given problem domain X then can defined by:

$$i^* = \arg \max_{i \in \{1, \dots, n\}} \max \{u_0(x) \mid x \in X, u_i(x) \geq v_i\},^3$$

where v_i denotes the reservation value of seller i .

To determine the outcome, we also need the second-best offer. Assuming all sellers follow the dominant strategy, the second-best offer $\hat{x}$ is given by

$$\hat{x} = \arg \max_{x \in \{x \mid u_i(x) \geq v_i, i \in \{1, \dots, n\} \setminus \{i^*\}\}} u_0(x).$$

The outcome then is the best possible for the winner i^* , given that it is at least as good for the buyer as the second-best offer $\hat{x}$, i.e.,

$$\omega = \arg \max_{x \in \{x \mid u_0(x) \geq u_0(\hat{x})\}} u_{i^*}(x).$$

Intuitively, this auction-like mechanism selects the best seller, because the sellers have a dominant strategy to submit their best offer, and the best of these is chosen by the mechanism in the first step. Moreover, the outcome selected is Pareto-efficient, because in the last step the winner maximizes its utility given a constraint on the utility for the buyer (and full knowledge of both preferences). The exact conditions and the proofs for efficiency and the dominant strategy equilibrium can be found in [8].

The main problem with a realistic implementation of the QVA is that the buyer needs to communicate all her preferences to all sellers. This is impractical for various reasons. Firstly, in many settings it is undesirable for the buyer to communicate all her preferences to all sellers, because the buyer may not want to disclose all details for strategic reasons. Secondly, this preference function can be a quite complex function over a large domain, which is difficult to communicate efficiently. Finally, a buyer may not even know the complete domain of agreements on forehand, even though she is able to rank any given subset of agreements. The latter holds for example when a government sends out a request for proposals to construct a bridge over a river within a given budget. It is impossible to list all possible types of bridges designers may come up with. But also in domains such as the super-computer domain, sellers usually come up with new options and alternatives in a negotiation process. If only the limited domain known by a buyer is used, the resulting outcomes will generally not be efficient. Therefore, in the complex multi-issue domains we consider in this paper, a standard ascending/descending auction, or the qualitative Vickrey auction discussed above cannot be used, because in such an auction the sellers require complete knowledge of the preferences of the buyer. In the next section we describe an approach based on negotiation that may be used to approximate the efficient outcome of such an auction and where there is no need to publicly announce the preferences of the buyer.

³We assume ties are broken by the buyer using a given ordering over the sellers.

6.4 Negotiation Protocols

In a QVA, all sellers propose an offer to the buyer. The buyer then determines which of the sellers has the winning offer. That seller is then allowed to change his offer to improve his utility value while taking into account that the buyer's utility value may not be lower than the second-best offer. In this setting, the dominant strategy for sellers in the first step is to propose an outcome that has a utility value equal to their reservation value with a maximal utility for the buyer. We could see this as an indication of the negotiation power of the buyer in a QVA. It means that sellers need to be aware that they are one out of potentially many other sellers that the buyer may reach an agreement with.

This negotiation power of the buyer explains why a QVA cannot simply be replaced by multiple bilateral negotiations based on e.g. an alternating offers protocol between the buyer and each of the sellers as this would not take into account that multiple sellers are contending for an agreement with the buyer. In [11] it was shown that a negotiation using the alternating offers protocol without any additional assumptions except for the fact that agents were able to learn opponent preferences does not result in a good approximation of the efficient outcome of the QVA.

In order to relax the constraint of the QVA that a buyer has to publicly announce its preferences, we propose three different negotiation protocols that take the negotiation power of the buyer into account. The first protocol we study tries to stay as close as possible to the QVA and imposes quite strict constraints on the moves of the negotiating parties. In fact, this protocol may be viewed as a variant of the QVA that does not require disclosing the preferences of the buyer. That is, this protocol consists of two negotiation rounds where in the first round sellers are constrained and required to propose offers that have a utility equal to their reservation value to the buyer and in the second negotiation round the buyer is constrained and required to propose offers that have a utility that is equal to that of the second-best outcome of the first round. Although this protocol is an improvement over the QVA in the sense that it does not require the public announcement of complete preferences, it still requires the negotiating parties to reveal their reservation value.

In order to remove the requirement to reveal reservation values, a second and a third protocol are studied that involve multiple negotiation rounds instead of just two rounds. The main idea is that in these future rounds sellers are provided an opportunity to outbid the winner of the previous round. The negotiation power of the buyer is represented in this protocol by the fact that negotiation continues over multiple rounds until no seller is willing to outbid the best outcome of the previous round (from the buyer's perspective). Both protocols are variants of this idea, where the second protocol requires the buyer to announce the winning bid at the end of each round and the third protocol only tells each player whether he or she is the winner at the end of each round.

6.4.1 A protocol based on two negotiation rounds

The first negotiation protocol consists of two rounds which closely match the QVA mechanism (see Algorithm 2 below for details). The buyer however is not required to announce his preferences. In the first round (step 1 of the algorithm) bilateral negotiation sessions are performed between the buyer and every potential seller. The idea is that in the first round negotiating parties try to learn a model $\tilde{u}(x)$ of each others' preferences, both in order to win the first round as well as to be able to perform well in the second round. Throughout the paper we use an alternating offers protocol [15] in the bilateral negotiation sessions. Furthermore, we assume that information about a negotiation session is not used in negotiation sessions with other sellers. At the end of the first round a winner (one of the sellers) is determined by the buyer. Then a second negotiation round (step 2) between the buyer and the winner is performed. Before starting this second round, however, the agreement between the *second-best* offer from one of the sellers and the buyer (from the perspective of the buyer) is revealed to all sellers. This is in particular useful for the winner who continues negotiation with the buyer. In the second round a final agreement between the winner and the buyer is established.

Algorithm 2 A protocol based on two negotiation rounds

1. Round 1: The buyer bilaterally negotiates with every seller.
 - (a) The buyer starts negotiation with each seller i .
 - (b) The seller proposes bids that have a utility equal to the seller's reservation value.
 2. Round 2: The buyer determines the winner and negotiates a final agreement with the winner.
 - (a) The buyer determines the seller with the highest agreement, as well as the second-best outcome.
 - (b) The winning seller and the buyer negotiate the final agreement. In this round the buyer is required to make offers with a utility equal to that of the second-best outcome of the previous round.
-

For this protocol to obtain efficient agreements, there are additional constraints on the negotiation strategies or procedure that the buyer and seller are required to use. In fact, the parties are required to propose offers that implement the steps of the QVA mechanism quite closely. In the first round all offers proposed by sellers are required to have a *utility equal to their reservation value* (see step 1.2.). This constraint is derived from the fact that the dominant strategy in the QVA for sellers is to propose such offers. In the second round all offers proposed by the buyer in the alternating-offers negotiation session are required to have a *utility equal to that of the second-best outcome of the first round* (see step 2.2.). This constraint is derived from the rule to compute the final outcome in the QVA. The intuition is that the buyer knows that an agreement with another seller of a certain quality can be reached. This should induce the winner to reduce the negotiation space it considers. An alternative way of putting this is that the winner of the first round is required to adjust its reservation value and increase it to the utility it associates with the

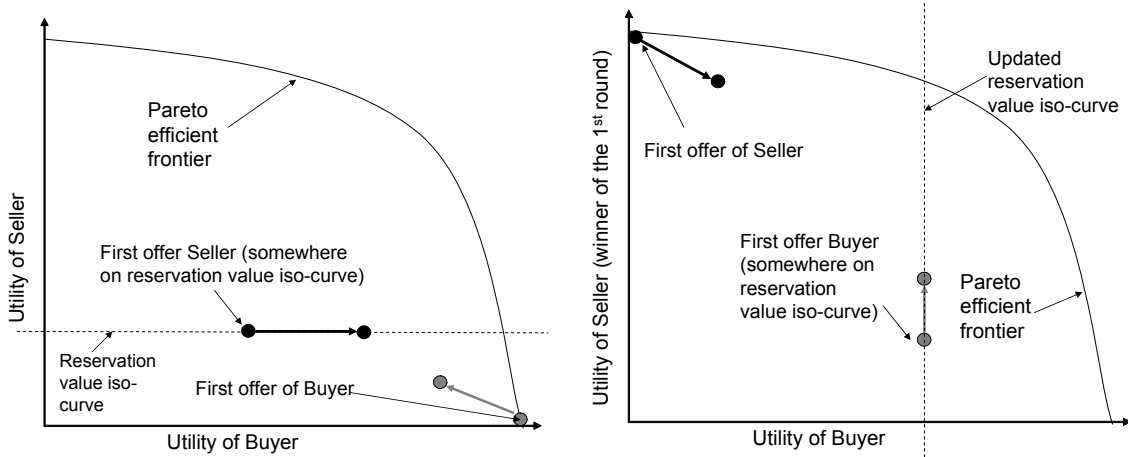


Figure 6.1: Negotiation moves in the first protocol: on the left-hand side the first round is illustrated where a seller stays on its reservation value iso-curve, and on the right-hand side it is shown how the buyer stays on the iso-utility curve of the utility of the second-best agreement of the first round.

second-best outcome as revealed by the buyer (if that outcome has a higher utility than its initial reservation value; otherwise, the seller would not change its reservation value).

Given that utilities are either fixed for the buyer or the seller, it is rational for the parties to try to propose the best agreement possible for the other party. In general this is the case since negotiators need to take into account that an offer needs to be reasonable for the other party in order to reach an agreement at all. In particular, this is the case for sellers in the first round, because they need to win in this round to go through to the second.

Given this setup, our hypothesis about the feasibility to approximate the mechanism outcome by means of negotiation is the following.

Hypothesis 1. *The outcome determined by the mechanism can be approximated by a negotiation setup in which: (i) the buyer does not reveal her preferences, (ii) the negotiating agents can learn an opponent's preference profile, and (iii) these agents use the negotiation procedure discussed above (Algorithm 2).*

Note that in the first round, the buyer is free to choose the offers he proposes. This makes it possible for the seller to learn the preferences the buyer has during this negotiation. Also note that as the seller is supposed to propose offers with a fixed utility value (equal to its reservation value) it is difficult if not impossible for the buyer to learn the preferences of sellers in this round. Figure 6.1 illustrates the first protocol and the constraints on the negotiation moves of the sellers in the first round and the buyer in the second round.

The major drawback of this first protocol is that there is no way to dictate or control the restrictions for bidding behavior of the sellers and the buyer. Either the buyer and the sellers have to trust each other that they comply with the negotiation protocol or a third party trusted by all agents has to be invited to control the bidding. There is, however, some incentive for the sellers, which can be derived from the similarity to the QVA where

proposing an offer at the reservation value is a dominant strategy. In the next section the need for such a trusted third party is removed.

6.4.2 Multiple negotiation rounds with multiple sellers

To remove the restrictions on the negotiation imposed in the first protocol, we introduce a protocol (see Algorithm 3 for details) that consists of multiple rounds of (parallel) bilateral negotiations between the buyer and the sellers. After each round r (step 2.2), the buyer communicates the *winning agreement* ω_i^r (where i is index of the winning seller) of round r to the sellers that did not win (i.e. they did not reach an agreement that was best from the buyer's perspective). All of the sellers then are provided with the opportunity to improve the agreement they reached with the buyer in a next round of negotiation sessions. A seller will do so if he can make an offer that has a utility value above his reservation value v_i , which he supposes has a higher utility to the buyer than the winning agreement of the last round. Negotiation is therefore assumed to resume for the seller in a next round starting with the agreement reached in the last round. This process continues until no seller (except for the winner) is prepared to negotiate in a next round to improve their last offer. The winning agreement of the last round then is the final agreement of the negotiation process. The details of this process are given below and are illustrated in Figure 6.2.

It is advantageous for a seller to understand the buyer's preferences in this process, because this can be used to reach an agreement that satisfies the buyer as best as possible while at the same time maximizing the utility for the seller itself. In particular, such an opponent model $\tilde{u}$ can be used to assess if an offer can be made that has the same utility value as the winning agreement from the point of view of the seller but that has a higher utility for the buyer. Only if such an offer cannot be made, an additional concession has to be made. Without the ability to learn an opponent model such an assessment cannot be made, and the seller will drop out of the negotiation process.

Figure 6.2 illustrates that the size of the negotiation space is decreased in every next round. This is explained by the fact that the buyer only accepts offers that improve the winning agreement reached in the previous round (see step 2.1). This process forces the final agreement closer to that of the reservation value of the sellers, in line with the dominant strategy sellers have in the QVA. We thus formulate the following hypothesis.

Hypothesis 2. *The agreement reached using the proposed negotiation protocol converges to that of the efficient outcome of the QVA, assuming the negotiating parties are able to learn the preferences of their opponent.*

The proposed negotiation protocol does not require the buyer to publicly announce his preferences. The protocol thus provides a realistic alternative for the QVA, that, given the hypothesis formulated above, can be used in settings where a buyer aims to reach an agreement with one out of multiple sellers. The process of reaching such an agreement is more complicated than that of the Vickrey auction but does not require publicly an-

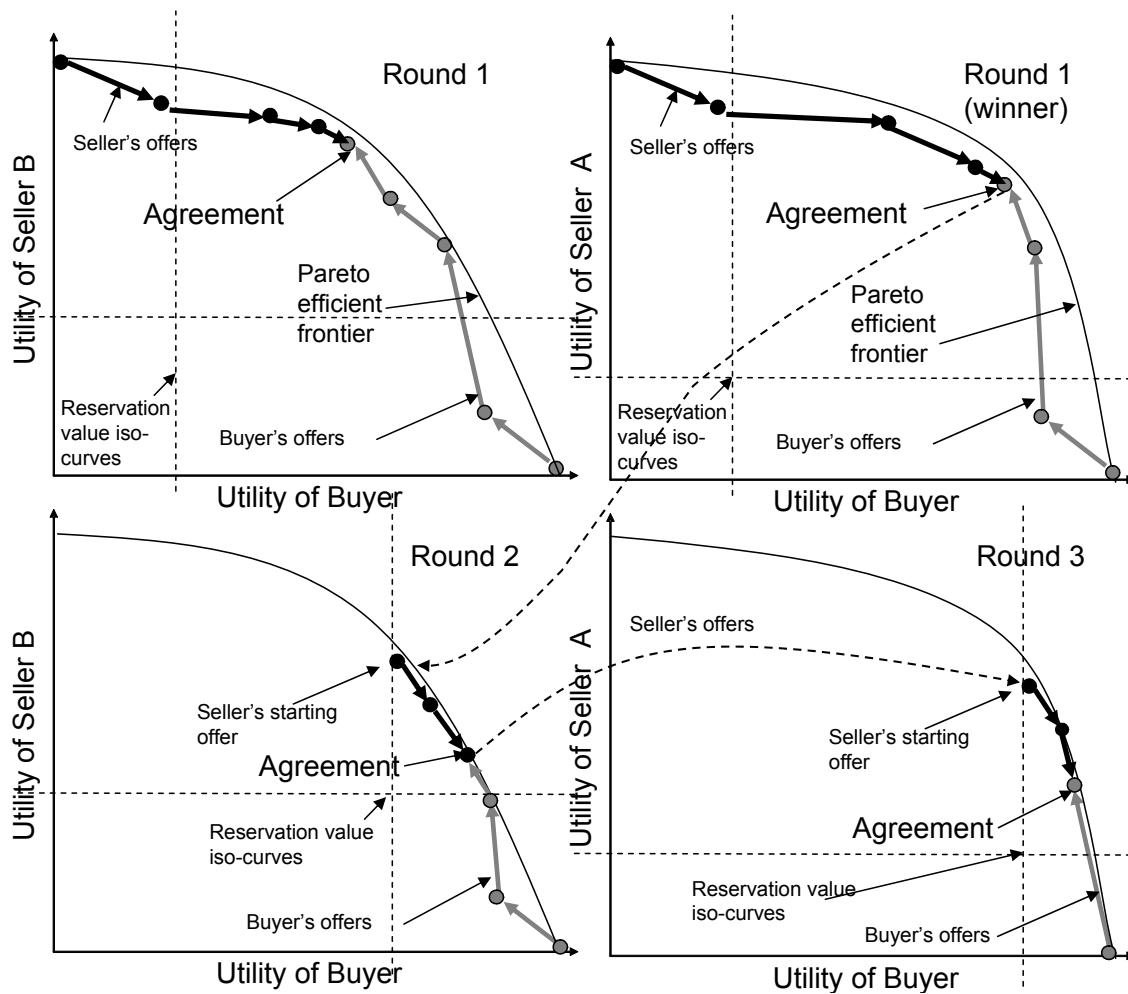


Figure 6.2: In round 2 Seller B aims to improve the agreement reached between the buyer and Seller A in round 1, and then in round 3 Seller A tries to improve upon this agreement.

nouncing the preferences of the buyer. Somehow the situation is reversed, however, as the protocol outlined above requires the public announcement of the winning agreement in every negotiation round (step 2.3). Instead of making the buyer's preferences public, in this case some information about the sellers' preferences is made public. We believe that this is not a prohibitive feature of the protocol as this only provides limited information to the sellers, but it still is interesting to investigate if this step in the protocol can be replaced by one that reveals even less information.

6.4.3 A variant without making intermediate agreements public

A similar protocol can also be applied without informing sellers about intermediate agreements. In this case, the buyer only indicates to a seller that it did not win in the last round.

Algorithm 3 A protocol with multiple negotiation rounds

1. Set the utility of ω^0 to be 0. Set the round number r to be 1.
 2. While no final agreement ω has been found, do the following.
 - (a) The buyer bilaterally negotiates an agreement with every seller i , accepting only offers with a utility at least as high as the utility of ω^{r-1} .
 - (b) If only one seller has reached an agreement, then the agreement of the previous round is the final agreement $\omega = \omega^{r-1}$.
 - (c) Otherwise, publicly announce the winning agreement ω^r .
 - (d) Start a new round, setting $r = r + 1$.
-

The winning agreement of the previous round thus can no longer be used as a reference point that needs to be improved upon from the buyer's point of view, and a seller instead continues negotiation in the next round with the agreement it reached itself in the previous round. Moreover, in the previous protocol where a winning agreement is made public, a seller can estimate – given the opponent model it learns during a negotiation session – how much it has to concede to improve that winning agreement. This is no longer possible in this second setup. However, it is required that when the negotiation protocol terminates and a final agreement is reached that this agreement is made public in order to allow sellers to verify that the buyer has not manipulated the process. Making only the final agreement public is sufficient for sellers that have a reasonable opponent model to assess whether the process has been fair, as there should be at least one seller that can make an offer with approximately the same utility to the buyer at his own reservation value.

Consequently, in this second variant the sellers have less information on how to outbid the winning seller of the previous round. Still, the buyer does have this information as it knows the winning agreement of the previous round and, therefore, would only accept offers of a seller that improve the winning agreement of the previous round. Given this, we formulate the following hypothesis concerning this variant.

Hypothesis 3. *The agreement reached without revealing the winning agreement in each round converges to that of the efficient outcome of the QVA, assuming the negotiating parties are able to learn the preferences of their opponent.*

As the sellers have less information in this third setup, they will have more difficulty in proposing offers that improve the winning agreement of previous rounds and more rounds may be needed to explore options to find such offers. We therefore formulate the following hypothesis about the number of rounds needed to reach a final agreement in the third variant compared to that needed in the second.

Hypothesis 4. *On average more rounds will be needed to reach a final winning agreement using the third setup than the second.*

We have argued that it is important that parties are able to learn opponent preferences. One question that remains is whether the sellers have an incentive to learn, or that they can achieve the same or even higher utility without learning.

Hypothesis 5. *An agent will be better off by learning the preferences of the opponent than without learning.*

To test this hypothesis we present some evidence where we compare the results of using a negotiation strategy that uses (Bayesian) learning to another strategy that does not.

6.5 Experimental Evaluation

In this section, we first discuss the design of the experimental setup and then present the obtained results. We present experimental results to evaluate how well each of the three multi-bilateral negotiation mechanisms approximate the QVA, although they do not make the buyer's preferences public. We also investigate the number of rounds required in the second and third protocol, and we investigate whether learning dominates not learning.

6.5.1 Experimental Design

The first experimental design choice concerns the number of sellers that participate in the negotiations. While the mechanism nor the protocol limit the number of sellers, in the experiments we use only two sellers with distinct preference profiles. This already simulates the competition between sellers since only the best and second-best can influence the outcome. Admittedly, when more sellers participate, the buyer may make a mistake in some of the rounds in finding the best seller, because it usually cannot perfectly learn the preferences of all sellers. However, in the multi-round protocols, such mistakes are easily repaired in future rounds. In addition, increasing the number of sellers reduces the expected difference between the best and second-best seller. This may give better results (regarding the deviation from the outcome of the QVA), because when accidentally the second-best is chosen as the best, the outcome will only marginally differ.

The second choice concerns the domain of negotiation. We have deliberately chosen a very generic domain and even relaxed natural constraints on this domain to further ensure generality. In the experiments we have used the so-called service-oriented negotiation domain taken from [5]. This domain consists of four issues that need to be settled, which represent the various attributes considered relevant with respect to the service offered, and include for example delivery time, quality, duration, and a penalty. Although we did use the generic four-issue structure of this domain we did not impose specific restrictions on the preferences such as that a lower penalty is always preferred by a seller as would be natural in this domain. As a result we have more variation in the preference profiles than one would typically expect in this domain. This variation in preference profiles ensures the relevance of our results for other domains as well.

For the experiments we have created a set of 12 preference profiles per role each, 12 for the buyer role and 12 for the seller role. Preference profiles were represented as piece-wise linear additive utility functions and each party in addition was assigned a reservation value. The remaining parameters such as the relative importance of an issue (weights),

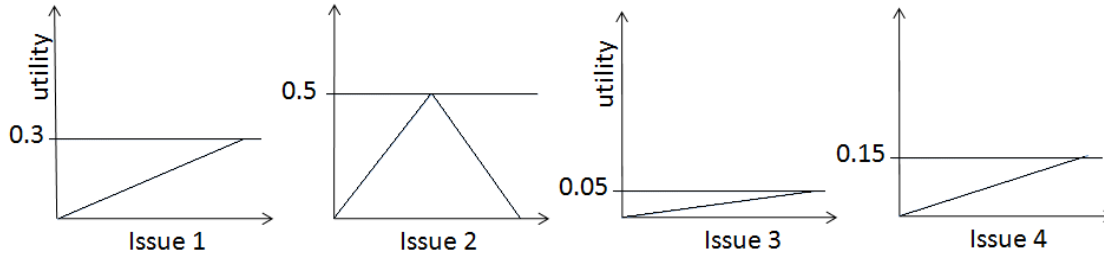


Figure 6.3: Example of a preference profile of a buyer with weights 0.30, 0.50, 0.05, and 0.15. Issues 1, 3, and 4 have “uphill” utility function, issue 2 has a “triangular” shape utility function.

the utility associated with the alternatives for each issue (called an evaluation function), and the reservation values are set as follows:

1. To model the relative importance of the value of the issues, two different sets of weights are used. One representing equal importance of all issues, using 0.25 as weight for each of the four issues, and a set of weights representing dominance of two issues over the other two, using the weights 0.30, 0.50, 0.05, and 0.15.
2. The utility associated with each of the alternatives associated with an issue were modeled by either a linear "uphill" function, a linear "downhill" function, or a combination of the two (resulting in a triangular shape). Two of the three types of evaluation functions are illustrated in Figure 6.3.
3. The reservation value for the buyer and sellers was set to either 0.3 or 0.6.

In Figure 6.3 an example of a preference profile for a buyer can be found. The relative scaling of the evaluation functions of the individual issue in the figure indicates its corresponding weight. The utility of a complete bid can be calculated by the summation of the utilities of individual issues.

Tables 6.1 and 6.2 show the predefined profiles that were created using variations of the three preference profile parameters defined above. The reservation value was varied with the preference profiles and set to either 0.3 and 0.6, and, as explained above, two weights vectors were associated with issues ($\langle 0.30, 0.50, 0.05, 0.15 \rangle$ and $\langle 0.25, 0.25, 0.25, 0.25 \rangle$). In a typical negotiation scenario it is normal to assume at least some level of opposition between the buyer's and the seller's preferences. To ensure this, evaluation functions for the issues 1, 3, and 4 of the buyer's profiles are set to the "uphill" type and the seller's evaluation functions for the issues 2, 3, and 4 are fixed to the "downhill" type. To vary the level of opposition between the buyer's and the seller's profiles the type of the evaluation function of the remaining issue is set to one of the three possible types "uphill", "downhill", and "triangle". These variations result in a total of $2 * 2 * 3 = 12$ possible profiles per role.

A sample of 50 different negotiation setups is created by means of a random selection out

Profile	w_1	w_2	w_3	w_4	$eval.fn$	v_i
Buyer1	0.25	0.25	0.25	0.25	uphill	0.3
Buyer2	0.30	0.50	0.05	0.15	uphill	0.3
Buyer3	0.25	0.25	0.25	0.25	downhill	0.3
Buyer4	0.30	0.50	0.05	0.15	downhill	0.3
Buyer5	0.25	0.25	0.25	0.25	triangle	0.3
Buyer6	0.30	0.50	0.05	0.15	triangle	0.3
Buyer7	0.25	0.25	0.25	0.25	uphill	0.6
Buyer8	0.30	0.50	0.05	0.15	uphill	0.6
Buyer9	0.25	0.25	0.25	0.25	downhill	0.6
Buyer10	0.30	0.50	0.05	0.15	downhill	0.6
Buyer11	0.25	0.25	0.25	0.25	triangle	0.6
Buyer12	0.30	0.50	0.05	0.15	triangle	0.6

Table 6.1: Predefined buyer profiles.

of the twelve profiles from Tables 6.1 and 6.2 for each of the three roles (one buyer, two sellers). Moreover, as a seller with a lower reservation value in such a setup has a higher chance of winning the first round (due to convexity of the Pareto efficient frontier), the sample is balanced such that in 80% of the cases the sellers have equal reservation values. To generate 20% of the negotiation setups where sellers have unequal reservation values, a complete set of all possible seller pairs with unequal reservation values is build. This set is used for the random selection of the negotiation setups. The rest of the sample (80%) of the seller profiles with equal reservation values was generated in a similar way.

Finally, a choice has to be made concerning the type of negotiating agent and the strategy that agent uses. As we have argued above, learning a preference profile is an important capability required when the preferences of the buyer are not publicly known. For this reason, we use an agent capable of learning a preference profile in a single negotiation session using Bayesian learning introduced in [9]. In the experiments, this negotiating agent builds a model of opponent preferences by learning a probability distribution over a set of hypotheses about the utility function of its opponent. In our case the agent has to learn the weights of issues and the corresponding evaluation functions. These structural assumptions make the learning task feasible.

We briefly explain the learning mechanism itself, for details please see [9]. During a negotiation session every time a new bid is received from the opponent the probability of each hypothesis about the opponent's utility function is updated using Bayes' rule. To be able to use Bayes' rule the conditional probability that the bid might have been proposed given a hypothesis is used. The utility of the bid according to the current hypotheses is computed and compared with a predicted utility based on the assumption that the opponent uses a concession-based tactic. This assumption is rational as in general any negotiator has to concede to reach an agreement. The details of the strategies in combination with the protocols as used in our simulation experiments can be found in Appendix A.

Profile	w_1	w_2	w_3	w_4	$eval.fn$	v_i
Seller1	0.50	0.30	0.15	0.05	uphill	0.3
Seller2	0.25	0.25	0.25	0.25	uphill	0.3
Seller3	0.50	0.30	0.15	0.05	downhill	0.3
Seller4	0.25	0.25	0.25	0.25	downhill	0.3
Seller5	0.50	0.30	0.15	0.05	triangle	0.3
Seller6	0.25	0.25	0.25	0.25	triangle	0.3
Seller7	0.50	0.30	0.15	0.05	uphill	0.6
Seller8	0.25	0.25	0.25	0.25	uphill	0.6
Seller9	0.50	0.30	0.15	0.05	downhill	0.6
Seller10	0.25	0.25	0.25	0.25	downhill	0.6
Seller11	0.50	0.30	0.15	0.05	triangle	0.6
Seller12	0.25	0.25	0.25	0.25	triangle	0.6

Table 6.2: Predefined seller profiles.

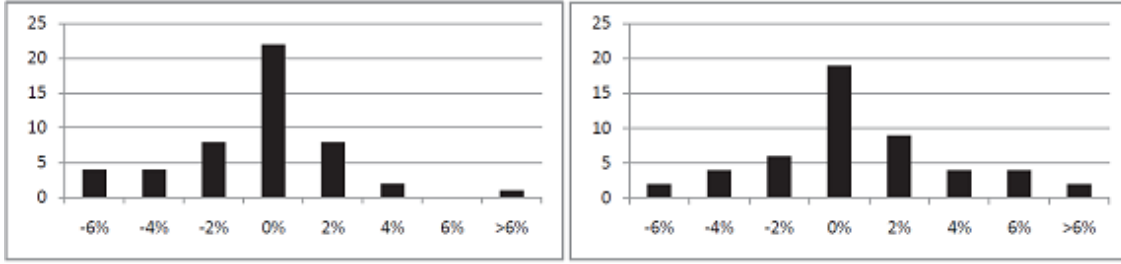


Figure 6.4: The distribution of the difference in utility for the buyer (left) and the seller (right) between the outcome of the first negotiation protocol and the outcome selected by the QVA.

6.5.2 First Negotiation Protocol

This first set of experiments should test our hypothesis that negotiating agents that use the first protocol and can learn a preference profile on the fly are able to approximate the outcome determined by the QVA mechanism quite well. For this, we study two results. Firstly, we compare the number of times the same winner is selected by the first multi-bilateral negotiation mechanism as by the QVA. Secondly, we study the differences in utility for both the winner and the buyer in case the same winner is selected.

First of all, in our experiments the winner defined by the QVA mechanism and the winner of the multi-lateral negotiation protocol in the experiments completely coincide. This means that in the first round (of this first negotiation protocol) the same seller is selected as a winner as in the QVA.

Next, consider the differences in the utility of the outcome for both the buyer and the winning seller, represented by the histograms in Figure 6.4. These results are obtained

in the second round of the protocol and show that in general the outcomes obtained via the negotiation protocol approximate those of the QVA mechanism. In 78% of the experiments the difference is less than 5%. The average difference in utility for the buyer between the efficient outcome of the QVA and the experimental results is only -0.09% and the standard deviation is 4%. Moreover, in 94% of the experiment the difference was not more than 10%. For the (winning) seller the differences are even smaller (0.01% on average with a standard deviation of 5%), indicating that overall the outcomes were good approximations. Moreover, some of the bigger deviations could be traced back to difficulties with learning an opponent's preference profile.

To summarize, these two observations indicate that there is no reason to conclude that the QVA and the first negotiation protocol are significantly different, supporting Hypothesis 1. The (small) difference in utility from the outcome selected by the QVA can be explained as follows. In this protocol all agents try to maximize the opponent's utility while staying above their reservation value. For this, the ability of an agent to learn the preferences of an opponent is a key factor in a successful approximation of the auction mechanism. First, the selection of the winning (as well as the second-best) offer mainly depends on the ability of a seller to learn the preference profile of the buyer, because otherwise acceptable offers that maximize the buyer utility cannot be found. Second, the utility of the winning seller in the final agreement is determined by the buyer's ability to learn the seller's preference profile, because otherwise the outcome will not be near the Pareto front of the winning seller and the buyer. The difference from the utility of the QVA outcome can thus be explained by approximation errors in the used learning method.

This first protocol requires sellers to make only offers at their reservation value in the first round, and in the second round it requires that the buyer makes offers at the value of the second-best offer in the first round (see an example of a negotiation trace in Figure 6.5). In many cases, imposing such requirements is unrealistic. In previous work [11] we have seen that simply relaxing these requirements does not result in outcomes that are still approximating the outcome of the QVA. In this paper we therefore propose a multi-round protocol, which is evaluated hereafter.

6.5.3 Second Negotiation Protocol

This second set of experiments tests the hypothesis that the second (multi-round) protocol approximates the outcome of the QVA. Figure 6.6 shows an example of a negotiation trace for the second negotiation protocol. As above, the winner defined by the QVA and the winner in the negotiation experiments coincide in all of the runs. Again the outcomes obtained by using the negotiation protocol are quite close to those determined by the mechanism. Figure 6.7 shows the histograms of the differences of the utility of the outcomes. The average difference with the buyer's utility for the QVA outcome is 0.01% (with a standard deviation of 1.5%). The utility of the sellers differs from their utility of the QVA outcome by -0.37% (with a standard deviation of 1.6%). According to the t-test the difference between the means of the utilities of the QVA outcome and

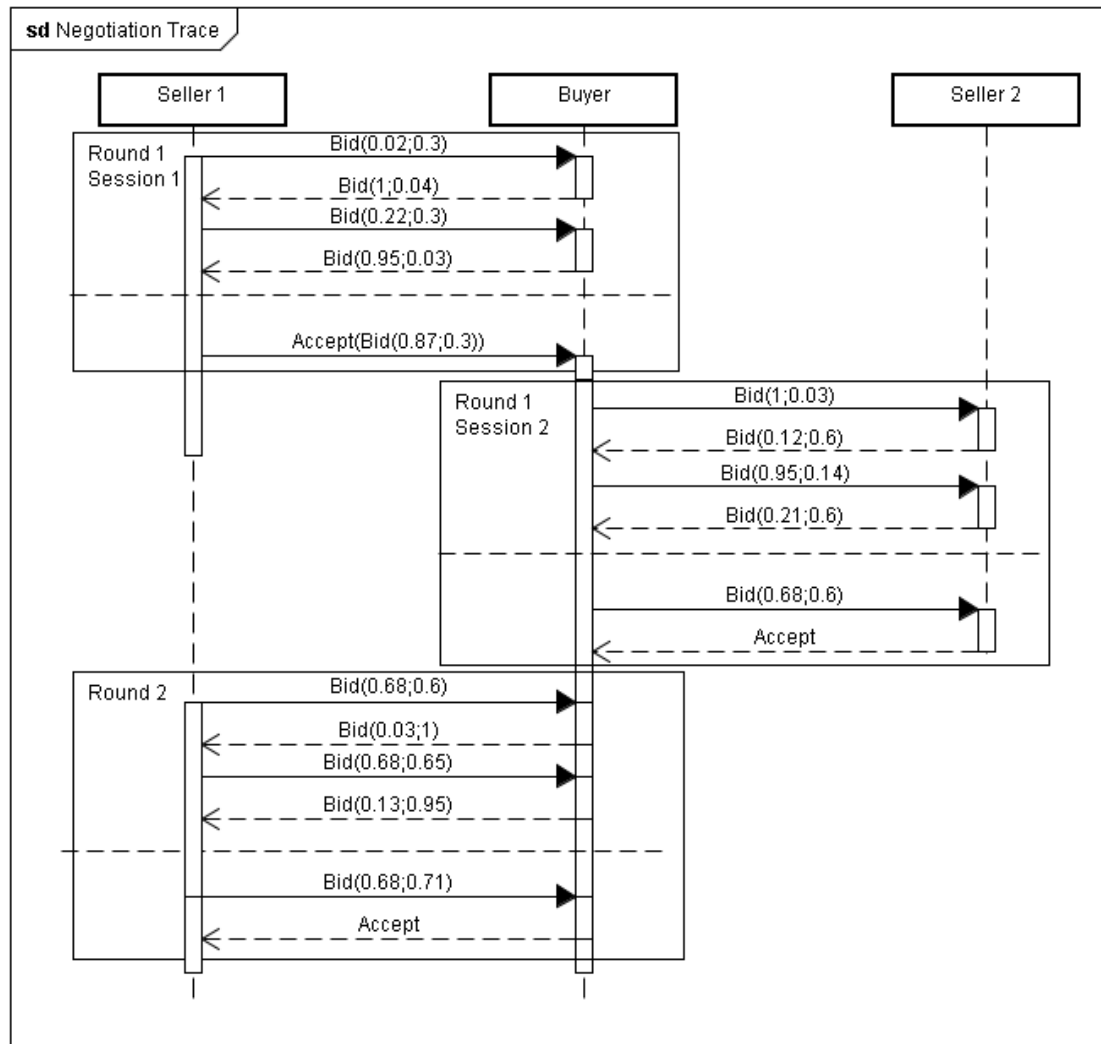


Figure 6.5: Example of a negotiation trace for the first negotiation protocol (each bid is denoted by the utility of the buyer and the seller). Seller 1 wins with an outcome with utility 0.68 for the buyer and 0.71 for the seller.

the experimental results are not only very small, but also insignificant (for the buyer: $t = 0.054, P(T < t) = 0.957$, for the seller: $t = 1.648, P(T < t) = 0.106$). This supports Hypothesis 2.

Moreover, as before, the difference of the experimental results and the utility of the outcome selected by the QVA can be explained by approximation errors in the learning method used.

6.5.4 Third Negotiation Protocol

Even the experimental results using the third negotiation protocol show only a small deviation from the outcome selected by the QVA. Figure 6.8 shows the histograms of the differences in utility for these outcomes. The average difference from the utility of the buyer's outcomes in the QVA is 1.39% (with a standard deviation of 2.2%). The utility of the sellers differ from the utility of the outcome in the QVA by -1.28% (with a standard deviation of 2.3%).

On average, the buyer gets a slightly better outcome in the proposed negotiation setup compared to the QVA outcome ($t = -3.8, P(T < t) = 0.00043$). This results in somewhat lower utilities for the sellers ($t = -3.9, P(T < t) = 0.00027$). These differences thus, although small, are significant. This can be explained by the fact that unlike in the auction mechanism, where the final agreement always corresponds to the reservation value of the second-best seller, in this last setup the sellers are not aware of each other's reservation value. Therefore, on the one hand, the deviation of the utility of the outcome is influenced by the size of the concessions made by the winning seller. As a result, the buyer can benefit from the seller's concessions. On the other hand, due to imperfection of the learned model of the opponent preferences, the second-best seller might drop out of the negotiation too early. The winning seller can benefit from this because no more concessions on her behalf are necessary. In such a case, the final agreement has a lower utility for the buyer. This relationship between the buyer's and the seller's utility of the final agreement can be observed in Figure 6.9. There we can see that one negotiating party can benefit from the underperformance of the other.

Even though the utilities of the outcome of this third negotiation protocol and the QVA are significantly different, they are still quite close. In addition, the same seller is always selected as a winner. We therefore conclude that also this third protocol is a reasonable approximation of the QVA, supporting our third hypothesis.

On average the number of the negotiation rounds in the third protocol is significantly higher than in the second setup (3.5 against 11.3 respectively, $t = -9.39, P(T < t) = 1.9 \cdot 10^{-12}$). Moreover, per round, using the third protocol almost two times more offers were made than in the second setup. That is, on average 50 offers were made in the second setup against 23 offers in the first one ($t = -14.4, P(T < t) = 4.14 \cdot 10^{-19}$). This confirms our fourth hypothesis.

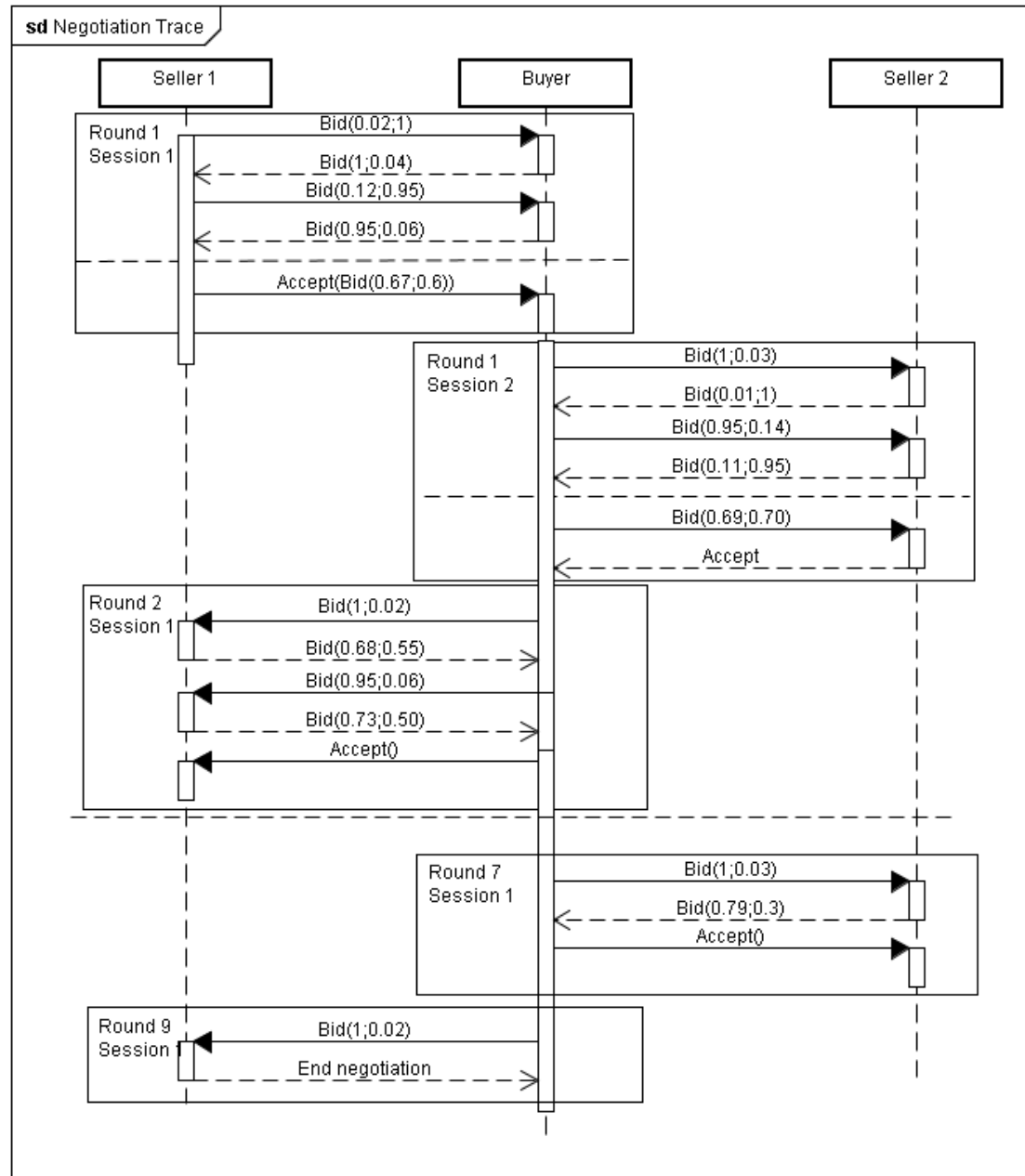


Figure 6.6: Example of a negotiation trace for the second negotiation protocol. Here Seller 2 wins negotiation with an outcome of (0.79, 0.3)

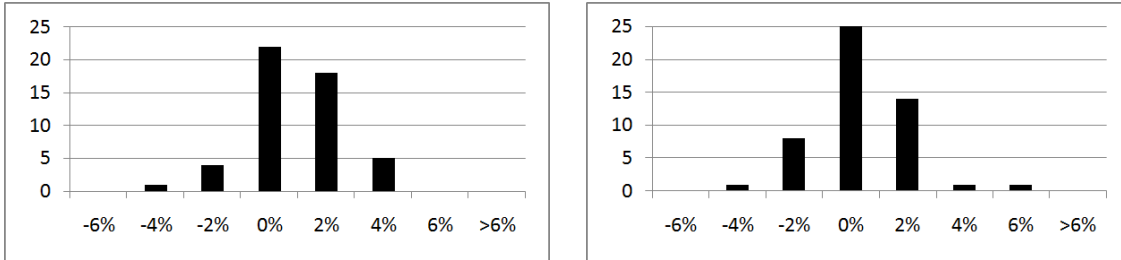


Figure 6.7: The distribution of the difference in utility for the buyer (left) and the seller (right) between the outcome of the second negotiation protocol and the outcome selected by the QVA.

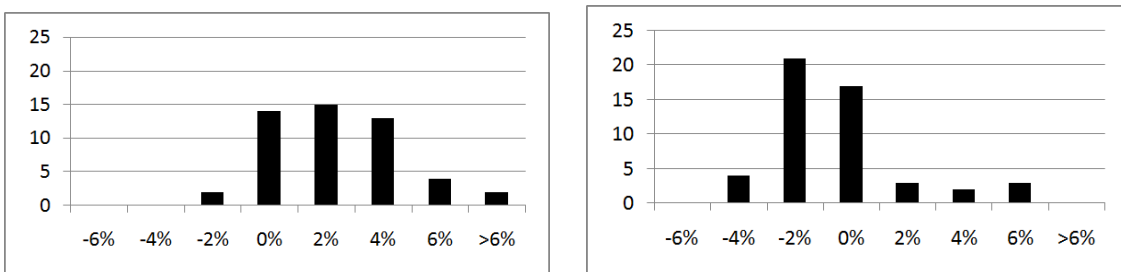


Figure 6.8: The distribution of the difference in utility for the buyer (left) and the seller (right) between the outcome of the third negotiation protocol and the outcome selected by the QVA.

Figure 6.9: The relation between the difference in utility for the buyer (vertical axis) and the seller (horizontal axis) for the second (left) and the third (right) negotiation protocol.

All these experiments are done on an Intel Pentium 4 3.0 GHz processor. For each experiment we measured the total run time of all rounds, including the Bayesian learning and the alternating-offers negotiation with two sellers in each round. For this third protocol, the run time is about 15.6 minutes on average, with a standard deviation of 23%. For the second protocol this is about 5.4 minutes on average with a standard deviation of 13%.

6.5.5 Negotiation Strategy

In the experiments discussed above we have shown that when all agents use the Bayesian learning strategy it is possible to approximate the outcome of the QVA. We also argued that learning is an essential part of any strategy that is able to realize outcomes similar to the auction. The question of quality of learning of the used learning technique was studied in [10] and, hence, is not covered in this paper. An important question that remains is whether such a strategy *dominates* other (non-learning) strategies. A strategy is said to *dominate* another strategy if it outperforms that strategy.

In order to (partially) answer this question we perform some additional experiments. These experiments are similar to the experimental setup described above but we replace one of the sellers with a seller that uses the Zero-Intelligence strategy [6]. The Zero Intelligence (ZI) strategy randomly proposes bids above its reservation value. On average, it is difficult for the ZI strategy to achieve a better agreement than its reservation value and any effective negotiation strategy is expected to outperform it. However, accidentally, the ZI strategy may make very smart moves and if a certain strategy always outperforms ZI, it may be concluded that this strategy also outperforms many other strategies. As before, 50 negotiation setups with two sellers and one buyer are used but this time for each of these setups two variants are run: (i) one where the first seller uses the ZI strategy and the second seller uses the Bayesian strategy, and, vice versa, (ii) where the first seller uses the Bayesian strategy and the other one uses the ZI strategy.

A summary of the experimental results of these 100 sessions is presented in Table 6.3. Every outcome of a negotiation session is classified into one of four possible cases, depending on the strategy of the winner (the first column), and whether the winner matches the winner in the QVA outcome (the second column). The third column provides the number of the negotiation sessions in each case. The deviation in the utility of the outcome for the winning seller and the buyer are given in the fourth and fifth column, respectively.

Most importantly these results show that there is not even a single negotiation session where a seller that loses in the QVA wins by switching from the Bayesian strategy to the ZI strategy (see the fourth row in the table). Moreover, a seller using the ZI strategy loses in 15 out of 50 runs, while it could have won using the Bayesian strategy. In addition, in cases where a seller wins in spite of using ZI, the utility is lower on average (1% better than the QVA outcome versus 6% better). These differences in utility are shown in more detail in Figure 6.10. Most of these differences can be explained by the fact that adding a ZI strategy complicates learning (the ZI strategy itself does not learn). It explains why on

Strategy of the winner in the experiments	Matches the QVA winner	Number of sessions	Deviation of the outcomes	
			Winning seller	Buyer
Bayesian	Yes	50	6% (always > 0%), st.dev = 3.5%	-6% (always < 0%), st.dev = 3.1%
	No	15	N/A (utilities of the sellers' should not be compared)	-8% st.dev = 6%
Zero-Intelligence	Yes	35	1% st.dev.=2.4%	-10% st.dev.=6.5%
	No	0		

Table 6.3: By changing to a ZI strategy, sellers do not win more often and do not receive a higher utility. When their competitor uses a ZI strategy, the winner using a Bayesian strategy stays the same and has a higher utility.

average the buyer's utility is significantly lower as it is impossible to learn anything from the offers proposed by a ZI strategy.

A second conclusion from these results is that sellers that follow the Bayesian strategy do not obtain worse outcomes when other sellers use the ZI strategy. This can be seen by looking closely at the results for the 50 cases where the seller following the Bayesian strategy wins both when using the negotiation protocol as well as according to the QVA. In those cases, the same seller is still a winner, and on average obtains a higher utility (6% difference from the utility of the QVA compared to 1.28% more utility on average in the previous section).

To conclude, the results show that outcome utilities of a seller that uses the Bayesian strategy do not get worse when the second-best seller switches to the ZI strategy, nor do they get better when the seller itself switches to the ZI strategy. Therefore, regardless of the choice of a strategy by the opponent, the most rational choice for the seller among these two is to stick with the Bayesian strategy. The large number of the conducted experiments with a seller that uses the ZI strategy provide a significant variation of the negotiation behavior of that seller. Given the fact that in all cases the ZI strategy did not perform better than the Bayesian strategy we can derive that the Bayesian strategy is a good choice for rational sellers regardless of the strategy of the other sellers.

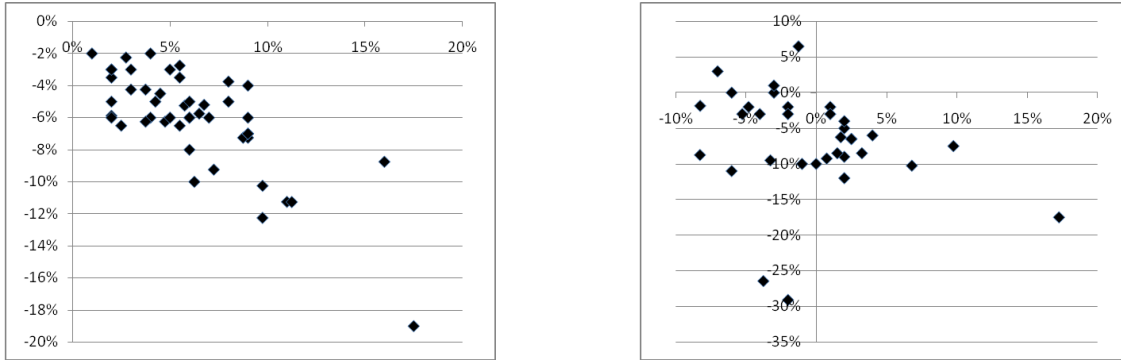


Figure 6.10: The relation between the difference in utility for the buyer (vertical axis) and the seller (horizontal axis) for the third protocol for the Bayesian (left, 50 cases) and the ZI (right, 35 cases) strategy.

6.6 Related work

The protocols presented in this paper relate to both auctions (because they select one winner among a set of sellers and the final agreements are efficient) as well as to negotiations (because the set of possible agreements is not agreed upon on forehand, and the parties do not know each other's preferences). In this section we therefore discuss both earlier work on (multi-attribute) auctions, as well as on multi-lateral negotiation, and we briefly discuss existing work on learning this context. Regarding the relation to auctions, the relation to the Vickrey auction and its generalization to multiple issues (possibly without money), the QVA, has already been discussed in the first two sections of this paper. The final two protocols, however, show stronger similarity to an auction that is even more familiar, i.e., the English auction.

Like in an English auction, in each round of the multiple bilateral negotiations, new agreements will only be accepted by the buyer if they are better than the best agreement in the previous round. When no other seller can make a better proposal, the process stops and the winning agreement is this best agreement. This process is very similar to an English auction, where bids are increased until all but one bidder stop bidding. The only difference is that in the setting discussed in this paper, the utility of the buyer is not known, while in a (reverse) English auction the item is fixed and the utility of the seller (buyer) is assumed to be linear in the price. This makes it very easy to come up with a better bid in an English auction, but quite hard to do so in our situation, where the sellers really need to learn the preferences of the buyer. The English auction is ex-post efficient, meaning that under the assumption that other sellers are rational as well, a seller can never do better than to follow a straightforward strategy of making an offer that is ranked slightly higher than the previous bid as long as this is above its own reservation value. We believe a similar result for our setting can be derived, supporting also theoretically that sellers can never do better than learning the buyer's preferences as well as possible, and then making concessions until either its offer is higher than the best offer of the previous round, or its

own reservation value has been reached (i.e., the fifth hypothesis).

In its basic form, the English auction, just like the Vickrey auction, is on one item that is completely described, but several generalizations of auctions have been proposed where some of the attributes of the item are left open for “negotiation”. However, in extant work the payments are always seen as a special attribute for which the preferences of the buyer and the sellers are related: a lower price for the seller means a worse outcome for the buyer. For example, [3] analyzed situations where a bid consists of a price and a quality attribute, and proposed both first-price and second-price sealed-bid (e.g., Vickrey) auction mechanisms. His work was extended by [4] for situations where the good is described by two attributes and a price. They analyzed the first-price sealed-bid and the English auction, and derived strategies for bids in a Bayesian-Nash equilibrium. In addition, they studied a setting where the buyer can also strategize, and they showed when and how much the buyer can profit from lying about its valuations of the different attributes.

Later work on iterative multi-attribute auctions has focused on a finite (discrete) domain with quasi-linear utility [16]. For this domain two related protocols are proposed. In Non-linear&Discrete (NLD) a reverse English (or Japanese) auction is held simultaneously for every combination of attribute values (called a bundle). In such an auction the price is dropped for each bundle until just one seller remains. The winning bundle is the one that maximizes the difference between the valuation of the buyer and the price. Straightforward bidding for sellers in this auction is defined by bidding the ask price for a bundle if that has still a positive utility. Straightforward bidding is shown to be an ex-post Nash equilibrium for the sellers and to result in an efficient outcome, maximizing the gains from trade (equal to the one-side VCG mechanism). On the buyer’s side, strategizing can bring a benefit of at most the marginal value to the economy contributed by the winner. In addition, for the setting where the utility of the buyer is additive over all attributes, NLD can be simplified. The mechanism Additive&Discrete (AD) does not hold an auction for every possible bundle, but just for every value (level) of each attribute separately.

Our iterative multi-bilateral negotiations with increasing utility for the buyer differ from this work in the following aspects. First, we consider piece-wise linear domains that may be continuous, which is a strict generalization of the finite domains. Second, we consider price just as one of the issues, which allows us to consider also utility functions that are not quasi-linear, but also makes it hard to define the gains from trade. We focus on Pareto-efficiency instead. Third, the mechanism differs in that we use bilateral negotiations between the buyer and each seller over all possible bundles. Such a negotiation usually involves only the exchange of a very limited subset of possible bundles, which is much more efficient compared to holding an auction for each possible bundle. Fourth, in these negotiations both the buyer and the sellers try to learn each other’s utility function in order to make better proposals. In general, there is not enough information to learn these perfectly, while in the finite setting at the end of the auction the utility function of one of the sides is completely known to the other party.

How much information is revealed exactly is an important topic for our future work. In [16] this is measured using the normalized size of the set of possible weights for the

issues, since in both NLD and AD a utility function is defined by the weights over the issues. For the sellers, this set is given by the constraints determined from the bids under the assumption that they follow the straightforward strategy. The size of the resulting convex set is approximated by a simple Monte Carlo algorithm.

We are aware of one other paper explicitly discussing the use of learning in the context of multilateral negotiation or multi-attribute auctions. In [1] the buyer learns the cost function of the sellers under the assumption that they all have a fixed form where only P parameters need to be determined. This can be exactly computed using inverse optimization after P auction rounds, where in each round, the buyer makes a different utility function public and assumes the sellers are bidding according to a straightforward strategy regarding the announced utility function. Only round $P + 1$ is for real. In that last round, the buyer constructs a utility function that maximizes its revenue. In the Bayesian learning approach used in this paper the buyer does not announce, nor change its utility function between rounds. If he did, the sellers would have more difficulty learning and in fact, we have seen that this usually reduces the buyer's utility. Moreover, as discussed above, in our work there is an incentive for the sellers to learn as well as possible and then use a straightforward concession strategy. In contrast, in the setting of [1], there is a clear incentive for the sellers to hide their utility in the first P rounds, to prevent the buyer to exploit them in the final round.

A multi-unit version of multi-attribute auctions is discussed in [20]. In their setting, two types of attributes are distinguished: those that relate to the seller (called bid attributes), and those that do not (called negotiable bid issues). Utility information regarding the negotiable bid issues is communicated to the sellers in the form of linear bonuses and penalties. Sellers indicate what the values of the issues are and how much they are willing to sell. The mechanism returns the current price, and the seller can then accept, reject, or change the submitted issues. The price is obtained by the mechanism using a straightforward strategy.

The extension to also deal with multiple units is straightforward, but significantly increases the applicability. We believe our mechanism can also easily be adapted to deal with multiple (identical) units. In [20] the utility function of the buyer is assumed to be a weighted sum of the negotiable bid issues. There is no assumption on the utility of the sellers, except that it is quasi-linear (since all issues are related to the price). Again, quasi-linearity is a restriction compared to our model. The most relevant difference from our approach, however, is the fact that the utility function of the buyer is communicated, except for a fixed discount that may be different for each seller, while in our approach we explicitly do not allow this.

Related work on negotiation mechanisms that deal with multiple players is reported in [13, 14, 17]. Our approach differs in at least two regards. First, our aim has been to reach an agreement that is as close as possible to an efficient agreement as obtained by the QVA. Second, we propose a new negotiation protocol that is based on several rounds of multiple standard bilateral negotiation sessions where all participants that lost in an earlier round are allowed to make a proposal that is better than the winning proposal of

the earlier round. Below we consider these existing negotiation approaches in a bit more detail.

Rahwan et al. [17] and later Nguyen and Jennings [14] have proposed a negotiation framework where the buyer negotiates with a number of sellers concurrently, and updates its reservation value in all other negotiation threads with the value of an agreement, whenever one is made. The latter work presents experimental results on the effect of a number of negotiation strategies in a setting where each utility function is a standard linear combination of the issues. It seems that in such a parallel setting the speed of the negotiation threads may influence the changes in reservation value of the buyer and thus the result. In our work this is resolved because there is always a next round until all sellers except one decide to end the negotiation.

Another line of work in this field includes an expectation about results obtained in other threads [13]. Like in the work discussed above, the reservation value for the buyer is set based on events in the other threads. The interesting extension here is that the reservation value can be set at the expected best offer in other threads, or even in future threads.

As a final topic to discuss here, we briefly review an empirical study of comparing an auction mechanism with a negotiation mechanism [12]. The aim of this study is to estimate the impact of trading mechanisms on the price of an agreement. The paper considers data of 216 trades that result from either using an auction or a negotiation as the trade mechanism. It is concluded that trade mechanisms can have an influence on the number of suppliers due to, i.e., costs associated with a particular mechanism. The results show that the choice of the trading mechanism does not influence the price of an agreement, however. This seems to contradict our results for the third protocol, where we see a (albeit small) difference in utility. We believe this can be attributed to imperfect learning, but we leave a thorough investigation of the cause of this difference for future work.

6.7 Conclusion

In general, negotiations facilitate the expression of agreements in greater detail than auctions do, making it possible to arrive at better win-win solutions between the buyer and the seller. However, (reverse) auctions on the other hand can guarantee that the deal is with the best seller, and some auctions, such as the English auction or the Vickrey auction, remove the need for bidders to strategize, making it a lot easier to participate (following a straightforward strategy).

Sandholm, among others, acknowledges this, and proposes a combinatorial auction that allows for as much details in bids as the buyers and sellers would find useful, a method he called expressive commerce [18]. However, even in that approach, the preferences of the buyer are quasi linear and need to be given on forehand to allow all participants to make successful bids.⁴ The main problem this paper deals with is the fact that the preferences of

⁴Minor changes in the preferences are allowed afterwards (scenario navigation), but may influence the

the buyer in a one-to-many multi-issue setting may not be given on forehand and may not even be quasi linear, which is required for most auction types. Even the QVA requires that the utility function of the buyer is made publicly known. The main contribution of this paper is the idea of combining a negotiation protocol with the ability of the parties to learn the preferences of the opponent, removing the need to make these preferences public. The protocol proposed introduces multiple negotiation rounds in which sellers that lost in the previous round are given an opportunity to improve their offers and possibly outbid the winner.

We have discussed three variants of multi-round negotiations to approximate an auction. We showed experimentally that both the outcomes of the two-round protocol as well as the second (multi-round) protocol are not significantly different from the Pareto-efficient best-seller outcome of the QVA. The results of the third set of experiments indicate that even if no information is made public until the end of the negotiation the protocol closely approximates the outcome of the QVA. The number of rounds needed to find the winning contract, however, is significantly higher in the third than that used in the second protocol. This can be explained by the fact that sellers have no information about the winning agreement of the previous negotiation round and have to make more offers to be able to explore the outcome space before they are able to outbid the winner. Our results thereby show that a trade-off needs to be made between revealing preference information and the average amount of time needed to complete the negotiation. Our final set of experiments show that Bayesian learning in combination with a concession strategy dominates a Zero Intelligence strategy with random offers. This supports our hypothesis that the Bayesian strategy is dominant. A full proof of this claim is left for our continued studies.

For other future work, we are interested in potential forms of manipulation that may be available to the buyer in the third protocol in case the process cannot be monitored by a trusted third party. If a buyer has complete knowledge about the winner to be, he could lie about an offer in an earlier round. This “second-highest offer” can then be chosen in such a way that the negotiation space of the final agreement will be very small, in favor of the buyer. Finally, we also want to study how to modify the ideas presented in this paper to make the protocols presented applicable to a broader range of real-world one-to-many multi-issue negotiations over complex domains where preferences cannot completely be made public in advance.

Bibliography

- [1] Beil D, Wein L (2003) An inverse-optimization-based auction mechanism to support a multiattribute RFQ process. *Management Science* pp 1529–1545
- [2] Bichler M, Lee J, Lee H, Chung J (2001) Absolute: An intelligent decision making framework for e-sourcing. In: *Proceedings of the 3rd International Workshop on*

efficiency.

Advanced Issues of E-commerce and Web-based Information Systems, San Jose, CA, pp 195–201

- [3] Che Y (1993) Design competition through multidimensional auctions. *RAND Journal of Economics* 24(4):668–680
- [4] David E, Azoulay-Schwartz R, Kraus S (2002) Protocols and strategies for automated multi-attribute auctions. In: *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM New York, pp 77–85
- [5] Faratin P, Sierra C, Jennings NR (2003) Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence* 142(2):205–237
- [6] Gode DK, Sunder S (1993) Allocative efficiency in markets with zero intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* 101(1):119–137
- [7] Harrenstein P, Mahr T, de Weerd MM (2008) A qualitative vickrey auction. In: *Endriss U, Paul W G (eds) Proceedings of the 2nd International Workshop on Computational Social Choice*, University of Liverpool, pp 289–301
- [8] Harrenstein P, de Weerd MM, Conitzer V (2009) A Qualitative Vickrey Auction. In: *Proceedings of the ACM Conference on Electronic Commerce*, ACM Press, pp 197–206
- [9] Hindriks KV, Tykhonov D (2008) Opponent modelling in automated multi-issue negotiation using bayesian learning. In: *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pp 331–338
- [10] Hindriks KV, Tykhonov D (2008) Towards a quality assessment method for learning preference profiles in negotiation. In: *Proceeding of Workshop on Agent-based Complex Automated Negotiations*, pp 64–71
- [11] Hindriks KV, Tykhonov D, de Weerd MM (2008) Approximating an auction mechanism by multi-issue negotiation. In: *Hindriks KV, Brinkman WP (eds) Proceedings of the 1st International Working Conference on Human Factors and Computational Models in Negotiation*, pp 33–38
- [12] Kjerstad E (2005) Auctions vs negotiations: A study of price differentials. *Econometrics and Health Economics* 14(12):1239–1251
- [13] Li C, Giampapa J, Sycara K (2004) Bilateral negotiation decisions with uncertain dynamic outside options. In: *Proceedings of 1st IEEE International Workshop on Electronic Contracting*, pp 54–61
- [14] Nguyen TD, Jennings N (2003) Concurrent bilateral negotiation in agent systems. In: *Proceedings of 14th International Workshop on Database and Expert Systems Applications*, pp 844–849
- [15] Osborne MJ, Rubinstein A (1994) *A Course in Game Theory*. The MIT Press

- [16] Parkes D, Kalagnanam J (2005) Models for iterative multiattribute procurement auctions. *Management Science* 51(3):435–451
- [17] Rahwan I, Kowalczyk R, Pham H (2002) Intelligent agents for automated one-to-many e-commerce negotiation. In: *Proceedings of 25th Australasian Conference on Computer Science*, Australian Computer Society, Inc., Darlinghurst, Australia, pp 197–204
- [18] Sandholm T (2007) Expressive commerce and its application to sourcing: How we conducted \$35 billion of generalized combinatorial auctions. *AI Magazine* 28(3):45–58
- [19] Teich JE, Wallenius H, Wallenius J, Koppius OR (2004) Emerging multiple issue e-auctions. *European Journal of Operational Research* 159(1):1–16
- [20] Teich JE, Wallenius H, Wallenius J, Zaitsev A (2006) A multi-attribute e-auction mechanism for procurement: Theoretical foundations. *European Journal of Operational Research* 175(1):90–100
- [21] Thomas CJ, Wilson BJ (2005) Verifiable offers and the relationship between auctions and multilateral negotiations. *The Economic Journal* 115(506):1016–1031
- [22] Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16(1):8–37

Appendix A

Instantiations of the negotiation protocols with a negotiation strategy

Below we give the complete description of the negotiation processes used in this paper. In this description, the following notation is used.

$u_0(x)$	utility of the buyer
$u_i(x)$	utility of the i -th seller
$\tilde{u}_0(x)$	model of the buyer's preferences learned by a seller
$\tilde{u}^{s_i}(x)$	model of the i -th seller's preferences learned by the buyer
ω_i	outcome of the first round in negotiation with i -th seller (Algorithm 2)
i^*	winner of the first round (Algorithm 2)
$\hat{\omega}$	the second-best outcome of the first round (Algorithm 2)
ω_i^r	outcome of the r -th round in negotiation with i -th seller (Algorithm 3)

Algorithm 2 A protocol based on two negotiation rounds instantiated with a negotiation strategy which uses Bayesian learning to model opponent's preferences

1. Round 1: The buyer bilaterally negotiates with every seller.
 - (a) The buyer starts negotiation with seller i with its best bid $x_0^b = \arg \max u_0(x)$ and $t = 1$.
 - (b) Seller i updates the buyer's preferences model $\tilde{u}_0(x)$ given the buyer's bid x_{t-1}^b .
 - (c) The seller searches for a bid that has utility of the seller's reservation value and maximizes the expected utility of the buyer: $x_t^{s_i} = \arg \max_{x \in \{x | u_i(x) = v_i\}} \tilde{u}_0(x)$
 - (d) If the found bid has not been proposed before then the seller sends it to the buyer, otherwise stop the bidding.
 - (e) If seller stopped bidding, use the latest seller's bid $x_t^{s_i}$ as an agreement ω^{s_i} and go to step 2.
 - (f) Otherwise, the buyer updates the seller's i preference model $\tilde{u}^{s_i}(x)$ given x^{s_i} .
 - (g) The buyer makes a concession in its own utility space $c(t)$ and sends a bid that maximizes the expected utility of the seller i given this concession: $x_t^b = \arg \max_{x \in \{x | u_0(x) = u_0(x_0^b) - c(t)\}} \tilde{u}^{s_i}(x)$
 - (h) Increase t and go to step 1.2.
 2. Round 2: The buyer determines the winner and negotiates a final agreement with the winner.
 - (a) The buyer determines the winner $i^* = \arg \max_{i \in \{1, \dots, n\}} u_0(\omega_i)$ and the second-best outcome $\hat{\omega} = \arg \max_{i=1, \dots, i^*-1, i^*+1, \dots, n} u_0(\omega_i)$.
 - (b) The winning seller i^* starts negotiation with its best bid $x_0^{s_{i^*}} = \arg \max u_{s_{i^*}}(x)$ and $t = 1$.
 - (c) The buyer updates the seller's preferences model $\tilde{u}_{i^*}(x)$ given the seller's bid $x_{t-1}^{s_{i^*}}$.
 - (d) The buyer searches for a bid that has utility of the second-best outcome and maximizes the expected utility of the seller: $x_t^b = \arg \max_{x \in \{x | u_0(x) = u_0(\hat{\omega})\}} \tilde{u}_{i^*}(x)$
 - (e) If the found bid has not been proposed before then the buyer sends it to the seller, otherwise stop the bidding and use the buyer's latest bid x_t^b as a final agreement ω . Exit.
 - (f) Otherwise, the seller updates the buyer's preference model $\tilde{u}^{s_1}(x)$ given x^{s_i} .
 - (g) The seller makes a concession in its own utility space $c(t)$ and send a bid that maximizes the expected utility of the buyer: $x_t^s = \arg \max_{x \in \{x | u_i(x) = u_i(x_0^{s_{i^*}}) - c(t)\}} \tilde{u}^b(x)$
 - (h) Increase t and go to step 2.3.
-

Algorithm 3 A protocol with multiple negotiation rounds instantiated with a negotiation strategy which uses Bayesian learning to model opponent's preferences

1. Set the utility of ω^0 to be 0. Set the round number r to be 1.
 2. While no final agreement ω has been found, do the following for every seller i that still participates.
 - (a) The buyer starts negotiation with seller i with its best bid $x_0^b = \arg \max u_0(x)$ and $t = 1$.
 - (b) While no agreement in round r has been reached:
 - i. Seller i updates the buyer's preferences model $\tilde{u}_0(x)$ given the buyer's bid x_{t-1}^b .
 - ii. Seller i determines the utility of the next bid $u_t^{s_i}$:
 - A. If $t = 1$, then set $u_t^{s_i}$ to be the utility of ω^{r-1} or $\max u_{s_i^*}(x)$ in case $r = 1$.
 - B. Otherwise, make a concession $u_t^{s_i} = u_{t-1}^{s_i} - c(t)$.
 - iii. If utility of $u_t^{s_i}$ is less than the reservation value v_i , withdraw from negotiation.
 - iv. Determine a bid that has utility $u_t^{s_i}$ and maximizes the expected utility of the buyer: $x_t^{s_i} = \arg \max_{x \in \{x | u_i(x) = u_t^{s_i}\}} \tilde{u}_0(x)$ and send this to the buyer.
 - v. The buyer updates the seller's preferences model $\tilde{u}_{s_i^*}(x)$ given the seller's bid $x_t^{s_i}$.
 - vi. If the buyer's utility $u_0(x_t^{s_i})$ is more than the utility of the best outcome of the previous round, accept the seller's offer, i.e., $\omega_i^r = x_t^{s_i}$.
 - vii. The buyer makes a concession $c(t)$ and determines a bid that maximizes the expected utility of the seller i : $x_t^b = \arg \max_{x \in \{x | u_0(x) = u_0(x_{t-1}^b) - c(t)\}} \tilde{u}^{s_i}(x)$
 - viii. If the utility of u_t^b is less than the utility of $u_t^{s_i}$, accept the seller's bid anyway.
 - ix. Otherwise, send u_t^b .
 - x. If the utility of the buyer's next bid $u_0(x_t^b)$ is less than its reservation value v_0 then send a bid with the reservation value: $x_t^b = \arg \max_{x \in \{x | u_0(x) = v_0\}} \tilde{u}^{s_i}(x)$.
 - (c) If only one seller has reached an agreement, then the agreement of the previous round is the final agreement $\omega = \omega^{r-1}$.
 - (d) Otherwise, publicly announce the winning agreement $\omega^r = \omega_{\arg \max_i \{u_0(\omega_i^r)\}}^r$.
 - (e) Start a new round, setting $r = r + 1$.
-

Chapter 7

Eliminating Issue Dependencies in Negotiation Domains

In multi-issue negotiations, issues may be negotiated independently or not. In the latter case, the utility associated with one issue depends on the value of another. Searching for good bids in a utility space based on multiple, dependent issues in general is intractable. Furthermore, existing negotiation that have been proven to be efficient for negotiation in domains with independent issues cannot be used in case of dependencies between issues. Tractable algorithms do exist for independent issue sets, so one idea is to eliminate the dependencies by approximating the more complex utility space with issue dependencies. Several techniques have been proposed to deal with this increased complexity, including, for example, introducing a mediator in the negotiation setting. In this paper, an alternative approach based on a weighted approximation technique to simplify the utility space is proposed. It is shown that an approximation may give reasonable results when some structural features of the negotiation domain and preference profile are exploited. The approximated utility spaces can be used by existing negotiation algorithm for negotiation domains without issues dependencies. Of course, there is a risk that approximation results in significantly different negotiation outcomes. Therefore, a checking procedure to mitigate this risk is introduced and show how to tune the parameters of this procedure to control the outcome deviation. These parameters can be used to balance computational cost and accuracy of negotiation outcome. Based on experimental results specific values for the parameters of the checking procedure that provide a good balance between computational costs and accuracy are proposed.

7.1 Introduction

Negotiation is a process by which a joint decision is made by two or more parties [17]. The parties first express contradictory demands and then move towards agreement by a process of concession making. Negotiation is an important method for agents to achieve

their own goals and to form cooperation agreements, see e.g. [1, 2, 21, 22]. Raiffa [19] explains how to set up a preference profile for each negotiator that can be used during negotiation to determine the utility of exchanged bids. For more information on utility and other game theoretic notions the reader is referred to e.g. [16]. Representing agent's preferences in terms of mathematical formulae expressing relationships between values of issues and the utility of bids allows the development of software support for negotiations. The complexity of these relationships determines the computational costs of the negotiation process. One way to avoid such computational costs is, as proposed in e.g. [10, 11], to build up profiles as combinations of independent and simple evaluation functions per issue. This approach corresponds to the way the average human tackles negotiation. Humans tend to simplify the structure of their preferences ([24]) and prefer to negotiate one issue at a time, which means that issues influence the utility of a bid independently from each other. Absence of issue dependencies allows for the use of efficient negotiation strategies.

A number of efficient negotiation strategies exist for negotiation domains with independent issues, e.g. see [10, 4, 5, 8]. The strategies try to find efficient offers, that is the offers close to Pareto efficient frontier assuming that an efficient search algorithm is available to them. This is true for the domains where issues are independent of each other. In some domains, however, issue dependencies influence the overall utility of a bid. In such cases it is no longer possible to negotiate one issue at a time and Klein et al in [12] argue that there is no efficient method that an agent can use to negotiate multiple issues, even if the agent tries to guess the opponent's profile. The authors propose to use a mediator who uses a computationally expensive evolutionary algorithm that can solve non-linear optimization tasks of high dimensionality. Bar-Yam [1] shows that in a multi-issue negotiation with issue dependencies the utility can only be described by non-linear functions of multiple issue variables.

Until now this approach is only applicable if the values of the different attributes in the domain are independent from each other. However, in some domains the issues are interdependent. The AMPO vs City negotiation case study presented in [18] has generally linear additive structure of preference profiles of the negotiating parties but still has some weak dependencies between two issues, e.g. the City negotiating party would gain additional "bonus" utility if officers with less than 5 years service as well as the officers with more than 5 years would get now increase in vacation. Non-linear utility functions and preference elicitation methods are studied in the multiple criteria decision making and multiattribute utility theory [3, 15, 25].

In this paper, a new approach to tackle the complexity problem of a utility space with issue dependencies, called WAID (Weighted Approximation for Issue Dependencies), is proposed. It is based on the following observations. First, not all bids are equally important for negotiation: there are some bids which are not acceptable for the agent or are too optimistic to be an outcome of the negotiation. In effect, it is possible to indicate an expected region of utility of the outcome. Second, in real life cases a profile can be modeled by utility functions that are far from "wild"; they have a structure that is far

from random. This paper proposes a bid search algorithm based on weighted averaging as a method to approximate complex utility functions with simpler functions that is based on these observations. Furthermore, the bid search algorithm provides a way to check the adequacy of the approximation by a measure of the introduced error. The search algorithm can be used in the existing negotiating strategies designed for negotiation domains without dependencies between issues.

The negotiation outcome, however, does not only depend on the preference profile but also on the process of negotiation itself. It is to be expected that the risk of obtaining a bad outcome due to the use of an approximation cannot be avoided completely even if the approximation is quite good. In this paper, the risk of a bad negotiation outcome when using an approximation of the agent's preference profile is analyzed. It turns out that in some domains this risk may still be unreasonably high. The results show that using the approximated space a bid might be proposed that in the original utility space would have a too low utility. The risk of such an erroneous bid can be quite high and, as a consequence, the risk of obtaining a bad negotiation outcome is significant. In order to control this risk, the process of negotiation should be investigated as well. We investigate a way to incorporate a checking procedure to control the risk of an erroneous bid in the negotiation algorithm itself. This paper presents a checking procedure able to control the risk of erroneous bids which can be incorporated in any negotiation algorithm.

Of course, this checking procedure introduces some additional computational costs. One of the main contributions of this paper is that it shows that a trade-off can be made between computational efficiency and approximation accuracy, which is directly related to the negotiation outcome. The parameters of the checking procedure allow the tuning of the negotiation algorithm to increase either the computational efficiency or decrease the risk of erroneous bids. Derived from experimental results, we propose specific values for these parameters that ensure a reasonable balance between computational costs and outcome deviation (in terms of utility) in many domains. Finally, we present experimental results that show that the approach of adding a checking procedure to the negotiation algorithm is scalable and allows an agent to negotiate about high-dimensional utility spaces.

The paper is organized as follows. The next section reports on the related work in the field. Section 7.3 provides a formal definition of utility spaces with dependencies between issues and gives a leading case study that is used throughout the paper to illustrate the method. Section 7.4 describes the approximation method for eliminating such dependencies. The theme of Section 7.4.5 is the bid search algorithm based on the approximation method and shows that by varying certain parameters of the method a trade-off can be made between outcome deviation and computational costs. Section 7.5 analyzes performance of the proposed bid search algorithm in an experimental setup. Finally, section 7.6 concludes the paper.

7.2 Related Work

In [12] authors propose a negotiation protocol based on the mediated single text negotiation [18]. The protocol is specifically designed to handle complex utility spaces with dependencies between issues. In the protocol, a mediator proposes an offer that is initially generated randomly. Each agent then votes to accept or reject the offer. In case when both agents vote to accept the offer, the mediator mutates the offer and the new offer is sent back to the agents. If at least one of the agents rejects the offer the mediator mutates the most recent mutually acceptable contract and sends it to the agents. This procedure is repeated for a fixed number of times. This protocol can be scaled to negotiations with more than two agents.

Two types of agent strategies are used in [12]: “hill-climbers” and “annealers”. The hill-climbers accept an offer from the mediator if it has high utility than the most recent mutually accepted offer. The annealers can accept contracts worse than the one that is mutually accepted with a given probability. The probability depends on the utility change between the contracts and the time to the negotiation deadline. The annealers are tuned in such a way that the probability of accepting worse contracts is higher in the beginning of a negotiation and decreases to zero when the negotiation time approaches the deadline.

In the experimental setup the annealers showed better negotiation results than the hill-climber due to the fact that the hill-climbers tend to get stuck in a local optimum while the annealers can explore utility space more extensively by accepting worse offers. However, in negotiation setting with mix of a hill-climber and an annealer the hill-climber takes advantage of the annealer by dragging him into its own local optimum.

In [13] the work of [12] was extended by introducing annealing into the mediator. In this protocol, the mediator uses annealing to pursue not only the mutually accepted offers but also the ones that are rejected by the agents. To increase the mediator’s efficiency the agents are required by the protocol to annotate their votes with a form of strength: weak/medium/strong accept (reject). This modification allows to overcome the problem of negotiation settings with a mix of the hill-climbers and the annealers. However, the protocol can be sensitive to the truthfulness of the agents in annotating the strength of their votes.

The utility spaces that are used in [12] and [13] are defined by means of constraints in a multi-dimensional space. Every constraint is defined on a range of values in every dimension and has a single utility value. The utility space is then a sum of utilities of all constraints satisfied for a specific offer. The utility spaces defined in such a way are characterized by a “bumpiness” of the utility, i.e. sharp increases and decreases of the utility and a high number of local optimums. Such negotiation circumstances do not seem to be common in negotiation practice (see [11] for a number of examples of utility functions used in negotiation and decision making case studies) where preferences have some structure that is far from such “wild” behaviour.

In [9] the idea of using a mediator for negotiations with issue dependencies proposed in

[12] is extended by introducing a more advanced negotiation protocol. In the protocol, the agents randomly sample their own utility spaces to find areas of high utility and share them as well as the utilities of the areas with a mediator. The mediator finds overlaps between the reported areas and selects the one with the highest social welfare. The main disadvantage of the proposed protocol is in the necessity of a trusted third-party that would play the role of mediator. The mediator has to be trusted in keeping the shared information about the agent's utility spaces secret and being honest to all agents in maximization of the social welfare when searching for the final negotiation outcome. Furthermore, in the proposed protocol the agent's have an incentive to lie about their utilities in order to get better negotiation outcome for themselves.

The problem of negotiation with dependencies between issues is also studied in the context of bundling items under negotiation. Utility of a bundle can be non-linear with respect to the presence of items in the bundle, that is utility of two items in a bundle has different utility than a sum of utilities of the two items individually. Robu et al. in [20] propose a graph-based technique to learn complex opponent's profiles. The authors propose an algorithm of exponential computational complexity for searching through a learned utility space of the opponent. The main interest in [20], however, is the scalability of a model for representing an opponent's profile which is different from the approach proposed here to simplify an agent's profile. The technique can be only applied to negotiation domains with binary issues that represent presence of a product in the bundle.

In [23] authors propose a negotiation algorithm for a shop negotiating with a customer about a bundle of products and a price. The strategy is able to handle non-linear utility functions of bundles. It tries to combine aggregated knowledge about preferences of a potential customer with a preference model of a customer that is learned during negotiation on-line. Unfortunately, the strategy is tested only on relatively small domains with up to ten products in a bundle (that is $2^{10} - 1 = 1023$ possible bundles).

An interesting approach to approximation of non-linear utility functions is proposed in [6]. The authors split the non-linear utility function in several intervals and then approximate those intervals with linear functions. This can significantly improve computational tractability of a bid search algorithm. Authors assume that the approximations as well as the corresponding approximation error are given and do not give any recommendations on how to build them. Furthermore, authors do not consider the case of dependencies between the issues and leave it for future work.

7.3 Utility of Interdependent Issues

The overall utility of a set of independent issues can be computed as a weighted sum of the values associated with each of the separate issues. As is common (see e.g. [10, 19]), an evaluation function is associated with each issue variable and the utility of a bid then

is computed by the following weighted sum of the issue evaluation functions:

$$u(x_1, \dots, x_n) = \sum_{i=1}^n w_i ev_i(x_i) \quad (7.1)$$

In equation 7.1, the (weighted) contribution of each issue to the overall utility only depends on the value associated with that issue and the contribution of a single issue can be modeled independently from any other issues. Evaluation functions for independent issues thus have exactly the same properties as the utility function associated with the bids that consist of multiple issues: it maps issue values on a closed interval $[0, 1]$. This setup can be used for issue values that are numeric (e.g., price, time) as well as for issue values taken from ordered, discrete sets (e.g., colors, brands).

Bid utility functions that are weighted sums of the contribution of single issue values to the overall utility cannot be used, however, for modeling dependencies between issues. The value of one issue may depend on that of another, thus influencing the utility of a bid that includes both issues. Dependencies between these issues give rise to a generalization of equation 7.1 to:

$$u(x_1, \dots, x_n) = \sum_{i=1}^n w_i ev_i(x_1, \dots, x_n) \quad (7.2)$$

The representation of a utility space with non-linear issue dependencies as in equation 7.2 is similar to the model proposed in [12]. The main difference is that instead of considering only binary issue values, we allow multi-valued, discrete, as well as continuous issue ranges.

The complexity of a utility function determines the computational complexity of the negotiation process. One of the main problems in dependent multi-issue negotiation is the computational complexity associated with searching for appropriate bids in the corresponding utility spaces. In case a utility function of multiple issues is non-linear in these issues, i.e. there are issue dependencies, finding a particular bid in the utility space is intractable. Computationally simple and efficient approaches covered in [14] mostly rely on the independence of issues to determine their next bid and are not applicable¹.

As an illustrative example of dependent issues, in this paper, we consider the negotiation of an employment contract where two important issues are at stake: the number of days that have to be worked and the number of days that childcare will be provided by an employer. In the example, the candidate employee additionally has to take into account a dependency between these two issues: working time (issue variable x_1) needs to be balanced with the time s/he needs to spend with his/her child (issue variable x_2). Assuming that the partner of the candidate is working too and can take responsibility for only part of

¹As we discuss below, however, the approach can be adapted by using exhaustive search through the utility space, but becomes intractable and in practice works only for small utility spaces.

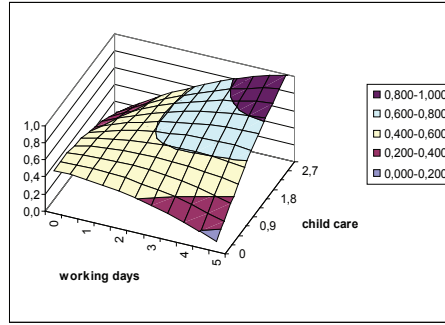


Figure 7.1: Utility space of the candidate employee with issue dependencies.

the childcare, the candidate has promised that s/he will take care of the child for at least 2 days, either by taking care in person, or by finding professional childcare. Thus the child care issue is really important and in case the employer proposes a contract for 5 days our candidate will try to negotiate a result which includes at least 2 days of childcare. In terms of utility, bids with 5 working days and less than 2 days of childcare have a low utility (e.g. $u(5, 0) \approx 0.1$, $u(5, 1) \approx 0.5$). In case the employer proposes a contract for only 4 days, the candidate will need to negotiate a result including only one day of childcare and a bid of 4 working days and one day of childcare has an acceptable utility value associated with it (e.g. $u(4, 0) \approx 0.25$, $u(4, 1) \approx 0.55$) though the candidate would prefer to work more. With respect to bids of the employer that require the candidate to work 3 days or less, there is no problem regarding the caretaking of the child. In that case, the childcare issue has much less influence on the value of the bid (e.g. $u(3, 0) \approx 0.35$, $u(3, 1) \approx 0.55$). Even in this relatively simple example, the values associated with each of the issues cannot be modelled independently and overall utility cannot be calculated using equation (7.1). The contribution of the childcare issue to overall utility depends on the number of working days associated with the other issue and vice versa in a way that introduces non-linear dependencies between the issues. Such non-linear dependencies can only be modelled by equation (7.2). To make the example concrete, the candidate's preferences are modelled using the following evaluation functions:

$$ev_1(x_1, x_2) = 0.01x_1^2 + 0.03x_1x_2 + 0.028x_2^2 \quad (7.3)$$

$$ev_2(x_1, x_2) = -0.04x_1^2 + 0.13x_1x_2 - 0.11x_2^2 + 1 \quad (7.4)$$

Figure 7.1 shows the utility space of the candidate employee defined by the evaluation functions (7.3) and (7.4) and weights $w_1 = w_2 = 0.5$.

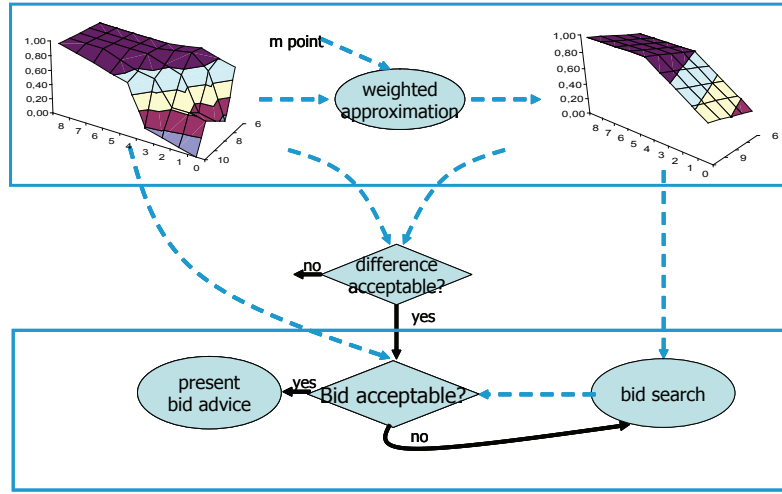


Figure 7.2: The WAID method overview.

7.4 Weighted Approximation Method

Due to the inherent computational complexity and the limited number of negotiation strategies that can be used to handle issue dependencies in negotiations, it would be beneficial to have methods that simplify the negotiation process of dependent issues without using a mediator. One particularly interesting option is to investigate the complexity of the utility space itself and try to eliminate the dependencies between issues. In case issue dependencies can be eliminated, various alternatives for efficient negotiation become available: Searching through the utility space of multi-issue bids becomes feasible and negotiation strategies for independent issues can be applied.

In this section, a method based on weighted approximation is proposed to eliminate issue dependencies (see Figure 7.2). It uses an averaging technique (weighted approximation) in which some general observations about negotiation have been integrated (the m -point and the weighting function that will be explained later) and which can take available knowledge about a negotiation domain into account. In particular, knowledge about the relative importance of bids and about outcomes which reasonably can be expected are part of the weighted averaging method.

Although elimination of issue dependencies implies a loss of information and accuracy with regard to utility, it is shown in this paper that if the influence of one issue on the associated value of another issue is “reasonable” (i.e., the utility space is not too wild) a good approximation of the complex utility space can be obtained.

The approximated utility can be used by a negotiation strategy to find an offer given some criteria, e.g. find an offer with a certain utility. The strategy can adopt a bid search algorithm proposed in this paper. Due to approximation error the found offer can have a significant deviation of utility in the original utility space. This deviation can be easily calculated due to the fact that calculating utility of an offer even using non-linear utility

space is computationally cheap. If the deviation is unacceptable the search algorithm would propose another offer. Such procedure is repeated until an offer with acceptable utility deviation is found.

The averaging technique proposed in this paper for eliminating dependencies is valid for utility spaces that have a certain “smooth” structure. The technique averages the values of bids close to each other. Therefore, utilities should not fluctuate too much from one bid to another within the proximity range set by the technique. In real life, common negotiations, this limitation on the applicability of the method is not seen as a problem considering that it is cognitively hard to make sense of wildly fluctuating utility spaces. As an indication, we think that the techniques are applicable to utility functions that can be modeled by polynomial functions of modest power. If the nature of the utility space is not clear, the applicability of the proposed techniques has to be tested for that case. A case study illustrates that the elimination of dependencies does not result in significant changes of the negotiation outcome. Additionally, a method for analyzing and assessing the difference between the original and approximated utility space is provided. This method analyze and assess the results can always be applied to arbitrary utility spaces.

Our main objective thus is to find and present a method for transforming a utility space $u(x_1, \dots, x_n)$ based on dependent issues that can be represented by equation (7.2) to a utility space $u'(x_1, \dots, x_n)$ without such dependencies that can be represented by equation (7.1). There exist various techniques to transform complex (utility) spaces with non-linear functional dependencies between variables to spaces which are linear combinations of functions in a single variable [18]. For our purposes, we are particularly interested in the linear separability of non-linear evaluation functions of dependent issues². The main idea is to transform a utility space $u(x_1, \dots, x_n)$ into an approximation $u'(x_1, \dots, x_n)$ of that space by approximating each of the evaluation functions $ev_i(x_1, \dots, x_n)$ by a function $ev'_i(x_i)$ in which the influence of the values of other issues x_j , $j \neq i$, on the associated value $ev_i(x_1, \dots, x_n)$ have been eliminated. Mathematically, the idea is to “average out” in a specific way the influence of other issues on a particular issue.

The WAID method takes as input a utility space based on non-linear issue dependencies (i.e. issues cannot be linearly separated and transforms it into a utility space that can be defined as a weighted sum of evaluation functions of single issues (i.e. issues are independent). The WAID method consists of the following steps:

1. As a first step, estimate the utility of an expected outcome that is reasonable (given available knowledge). This estimate is called the “*m*-point” and is used to define a region of utility space where the actual outcome is expected to be.
2. Select a type of weighting function. The selection of a weighting function is based on the amount of uncertainty about the estimated *m*-point (expected outcome) in the previous step.

²In geometry, when two sets of points in a two-dimensional graph can be completely separated by a single line, they are said to be linearly separable. In general, two groups are linearly separable in n -dimensional space if they can be separated by an $n - 1$ dimensional hyperplane.

3. Calculate an approximation of the original utility space based on non-linear issue dependencies using the m -point and the weighting function determined in the previous step. The result of this step is a utility space that can be defined as a weighted sum of evaluations of independent issues (a function of the form of equation (7.1))³.
4. Perform an analysis of the difference of the original and approximated utility space by means of a Δ -function to assess the range of the error for any given utility level. In this final step, based on the assessment, thresholds for breaking off the negotiation or accepting opponent's bids can be reconsidered.

Finally, the results of the WAID method can be used in combination with a particular negotiation strategy. In section 7.5.1, we study the results of using an approximated utility space for the child care example in a negotiation strategy and compare the results with an approach based on the original utility space. The sections below explain each of the steps in more detail and illustrate how these steps achieve the objective of eliminating issue dependencies.

7.4.1 Estimate an Expected Outcome

Any approach based on using uniform arithmetical averaging methods has the effect of discarding information uniformly. Such an approach does not take the final goal of negotiation into consideration: the negotiation outcome. A uniform averaging method is indifferent to the fact that even before negotiation starts it can be assumed that certain regions of the utility space are more relevant to the negotiation than others. Some general observations about the structure of utility spaces that can be associated with negotiations taken from actual practice provide additional insight that can be used to increase the effectiveness of an approximation technique.

Consider, to make clear what we mean, a worst case scenario in which two agents A and B associate completely opposite utilities with bids. In other words, what is valuable for agent A is of no value for agent B. Formally, we can express this opposition in terms of utility functions as follows:

$$u_A(x_1, \dots, x_n) = 1 - u_B(x_1, \dots, x_n) \quad (7.5)$$

Given these utility functions, it is easy to see that the Nash product is 0.25 with associated utility values $u_A(x_1, \dots, x_n) = u_B(x_1, \dots, x_n) = 0.5$ and the same point within the utility space is an efficient negotiation outcome when using Kalai-Smorodinsky criteria, that is, a Pareto-optimal outcome with equal utilities for both parties. Assuming such opposite interests, none of the agents would ever accept a bid which has a utility below 0.5.

³In the more general case of more than two issues, an evaluation function may depend on more than two issues and one of those issues has to be selected to be separated from the other issues.

Typically, however, negotiations do not fit such worst case scenarios and there is something to gain for both parties. Formally, this means that there exist acceptable negotiation outcomes, i.e. bids, with associated utilities that are higher than 0.5. In such cases, the utility spaces of the negotiating opponents are not completely opposite as expressed by (7.5). This line of reasoning makes clear that in general we may assume that the expected outcome of the negotiation is located somewhere in the open utility interval $(0.5; 1)$ and this region in the utility space is generally of more importance in a negotiation.

It follows from the previous considerations that some regions within the utility space are more important for obtaining a good negotiation outcome than others and in the WAID method proposed should be approximated as good as is possible. As a first step to identify these regions, an agent can estimate an expected outcome which would identify with some probability one of the more relevant points in the utility space. We call this point the “*m*-point”.

An agent will be able to estimate an expected outcome with reasonable exactness only if it has some knowledge about the opponent’s profile. In that case, as we illustrate below, the *m*-point can be computed in two steps. But even if an agent lacks any information whatsoever about its opponent an *m*-point can be based on considerations of the agent’s own utility space. In the latter case, we propose that the *m*-point can be identified with the average of the break-off point (an agent breaks off a negotiation in case any utility with a lower utility is proposed) and the maximum utility in the utility space. In the childcare example, the break-off point equals 0.37, which is equal to the minimum utility that still satisfies the candidate employee’s childcare constraint.

A second, more informed method to determine an expected outcome can be used when the agent does have some information, e.g. based on previous experience, concerning the opponent’s profile. In the childcare example, assuming that the employer will take the child care request seriously into consideration, but will try to minimize his contribution in this regard, bids with 1-2 child care days are reasonable to expect. Additionally, it may be more or less certain that the employer prefers the employee to work as much as possible and that these issues are independent from the other. Then, as an estimated model of the opponent’s profile, the following evaluation functions can be used, which, using equal weights of 0.5, result in the utility space depicted in figure 7.3:

$$ev_1(x_1) = x_1/5 \quad (7.6)$$

$$ev_2(x_2) = (3 - x_2)/3 \quad (7.7)$$

An estimate of the expected outcome can now be computed from the agent’s own utility space and the educated guess of the opponent’s utility space using Kalai-Smorodinsky criteria, which ensures that a Pareto-optimal outcome is selected and the expected outcome is not strongly biased in favour of either one of the parties (see figure 7.3). Calculating the utility in our example yields $m = 0.74$. This estimate may still be quite uncertain, but

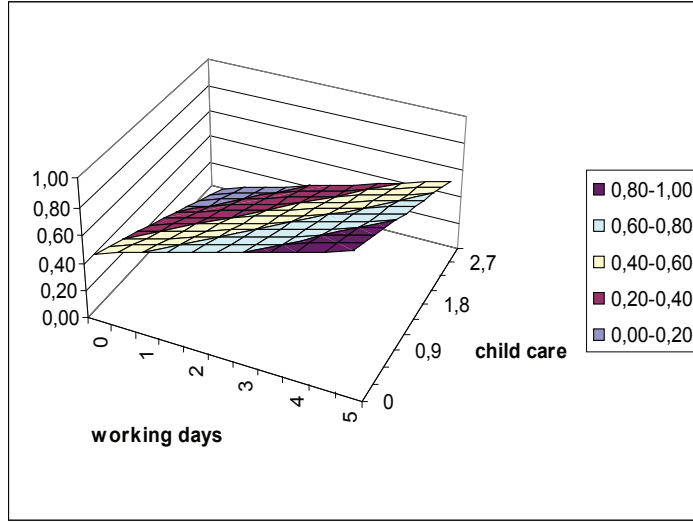


Figure 7.3: Utility space of the candidate employee with issue dependencies.

we will discuss this issue more extensively below. The estimated outcome only defines one parameter of the approach.

7.4.2 Select Weighting Function

As discussed above, not all points within the utility space are equally important for obtaining a good negotiation outcome. To take into account the relative importance of certain regions within the utility space, we introduce a weighting function associating a weight with each point (its “importance”) in the utility space. In general, there are two useful considerations that can be made which provide clues for constructing an appropriate weighting function.

The first consideration is that a certain range of utility values are of particular interest in the negotiation. Also, certain bids may be more “appropriate” than others in a negotiation. As an example, bids with utility values below a break-off point are less significant than other bids and do not have to be approximated as well as others. In the childcare example, provided with the relevant domain knowledge, it is moreover unreasonable for our candidate employee to propose to do no work and at the same time to request 5 childcare days.

The first consideration concerning the approximation of the utility space can be given a formal interpretation by associating the highest weight with the expected outcome (the “*m*-point” identified above, located within the $(0.5; 1)$ interval).

The second consideration is the fact that an agent may be more or less uncertain about its estimate of the utility of the negotiation outcome. To take this into account, we propose to use two different functions depending on the level of uncertainty that the agent has

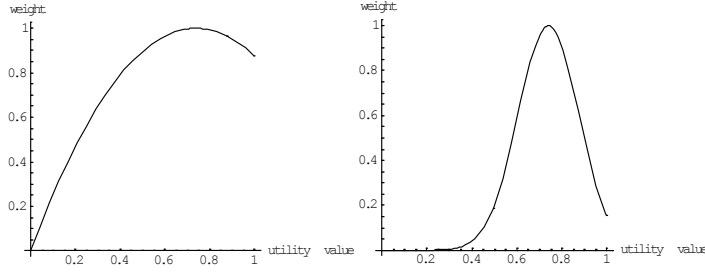


Figure 7.4: Example of ψ function for $m = 0.74$.

about the estimate of the m -parameter. In case the agent does not have information about the opponent, nor any past experience with the particular negotiation domain and is quite uncertain about the most probable outcome, a relatively broad range of utility values around the expected outcome should be assigned a high weight. As a consequence, bids in a rather wide neighborhood of the m -point are equally important for the negotiation and only extreme points (with utilities close to one or zero) do not have to be approximated very accurately. Given a relatively large uncertainty, we propose to use a polynomial function of the second order, which is rather flat near the m -point and declines closer to the extreme utilities (see figure 7.4 (left)). The corresponding weighting function ψ then can be computed as follows:

$$\psi(x_1, \dots, x_n) = \frac{2}{m}u(x_1, \dots, x_n) - \frac{1}{m^2}u^2(x_1, \dots, x_n) \quad (7.8)$$

In the case the agent is reasonably certain about the estimate, for example, when the most probable region of the negotiation outcome is well defined on the basis of domain knowledge, knowledge about the opponent or experience gained in previous negotiations, a weighting function with a stronger differentiation of utilities values can be used. In that case, a Gaussian function that is defined in terms of a maximum point m and spread σ can be used that assigns high weights only to bids with a utility close to the expected outcome m (see figure 7.4 (right)):

$$\psi(x_1, \dots, x_n) = e^{-\frac{(u(x_1, \dots, x_n) - m)^2}{\sigma^2}} \quad (7.9)$$

The spread parameter σ provides an indication of the agent's certainty about expected outcome. In both cases, the m -parameter represents the expected outcome and is a point in the interval $(0.5; 1)$; ψ assigns the m -point the maximal weight of 1.0.

In our example, an educated guess of the opponent's profile could be made and therefore a Gaussian weighting function is selected and a value for the "spread" σ needs to be determined. To this end, we use the 3σ rule (or "Empirical rule"), which says that (most

likely) 99,7% of all outcomes will be in the interval $(m - 3\sigma, m + 3\sigma)$, which gives us $\sigma = (0.37 + 0.74)/(2 * 3) = 0.19$.

7.4.3 Compute Approximation of Utility Space

Using the weighting function ψ a weighted approximation technique can be defined. The weighted approximation technique proposed here first multiplies each evaluation value with its corresponding weight and then averages the resulting space by integration. In the equation below, a function ω is introduced instead of ψ since the weighting must be normalized over the interval of integration. The range of integration is identical to the range of the integrated issue⁴. In case a negotiation involves n issues with interdependencies between these issues, and evaluation functions $ev_i(x_1, x_2, \dots, x_N)$ for the i th issue are given:

$$ev'_i(x_i) = \frac{\int_V \psi_i(x_1, \dots, x_n) ev_i(x_1, \dots, x_n) dV}{\int_V \psi_i(x_1, \dots, x_n)} \quad (7.10)$$

Here V is a volume of $n - 1$ dimensionality build on the dimensions $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$. Of course, not all issues have to depend on all others. The approximation technique can be applied sequentially for each evaluation function in the negotiation setup, which involves dependencies between issues.

As an illustration, we apply the weighted averaging technique to our employment contract negotiation. Figure 7.5 shows the ψ -functions for the original utility space using a polynomial function (7.8) for the left chart and a Gaussian function (7.9) for the right one. The flat section in the middle of the left chart represents a rather wide neighborhood of the m -point: this corresponds to the expected outcome and weights in its neighborhood are high. Outside this region the weighting function slowly declines to zero. For the Gaussian function (right chart) we obtain a different picture: the function has high values (close to 1) for the small band of bids with utility values close to the m -point and declines rapidly for the remainder of the utility space.

We apply expression (7.10) to the evaluation functions of our employment contract negotiation example to derive an approximated utility space without interdependencies from the original utility space. Figure 7.6 shows the original (top) and approximated utility spaces obtained by approximation with a polynomial weighting function (left) and obtained by using a Gaussian weighting function (right).

The utility spaces obtained by approximation with the polynomial and Gaussian weighting functions have a similar structure. However, the Gaussian weighting function due to its stronger utility discrimination power makes it more precise in the vicinity of the m -point. This is explained in more detail in the next section.

⁴If the issue has discrete values, integration simply means summation over all these values.

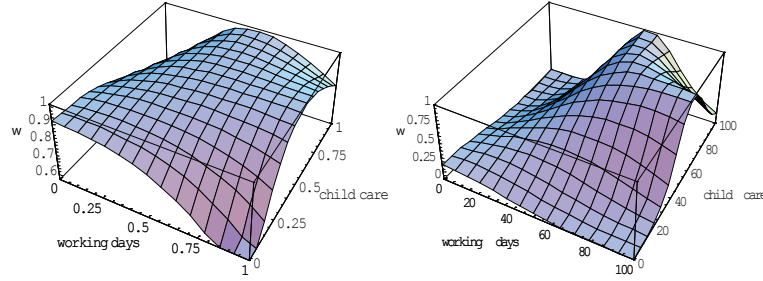


Figure 7.5: Examples of ψ -functions for the employee's utility space: (left) polynomial function with $m = 0.74$; (right) Gaussian function with $m = 0.74$ and $\sigma = 0.19$.

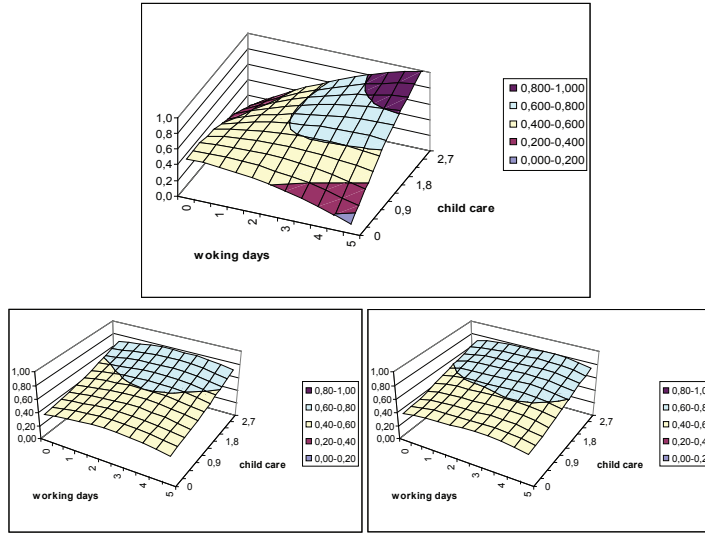


Figure 7.6: The original utility space (top) and new utility spaces of the employee obtained by (left) the weighted averaging method using a polynomial weighting function with $m = 0.74$ and (right) the Gaussian weighting function with $m = 0.74$, $\sigma = 0.19$.

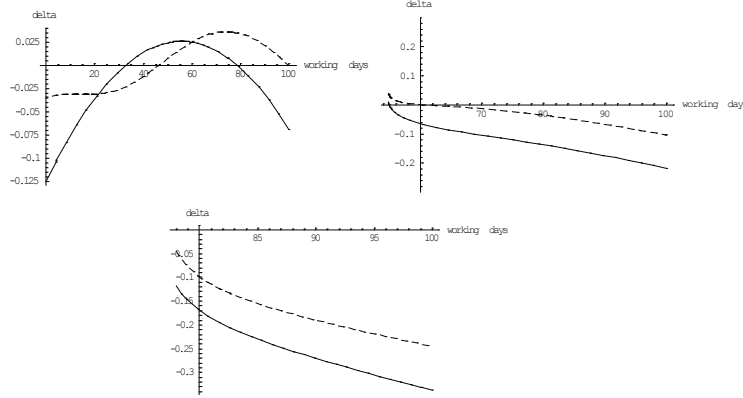


Figure 7.7: Graphs depicting values of the Δ functions for utility equal to (left) - 0.5, (right) - 0.7, (bottom) - 0.9 in the original space based on a polynomial weighting function (solid line), and a Gaussian weighting function (dashed line).

7.4.4 Analyze Difference Δ with Original Utility Space

The technique presented approximates the original utility space and consequently, introduces an error in the utility associated with bids. To obtain a measure for the distance of the values of bids in the original utility space compared to the bids in the approximated utility space, a difference function Δ can be defined as follows:

$$\Delta(x_1, \dots, x_i) = |u(x_1, \dots, x_N) - u'(x_1, \dots, x_N)| \quad (7.11)$$

As is to be expected, the Δ -values for the approximation using the Gaussian weighting function shift the utility considerably for some bids. For certain bids in the childcare example, the difference is almost 0.5. However, this only is the case for bids that are unreasonable and are not relevant for reaching a negotiation outcome. In particular, this shift in utility occurs for bids that involve more days of child care than working days. Approximations of the utility of bids that are close to the m -point are very good and close to zero.

To see the effect of the weighted averaging method near the m -point we take a section in the original utility space for the m -point ($m = 0.74$ for our negotiation example). By fixing the utility to 0.74, an expression can be obtained for the value of one of the issues as a function of another one:

$$u(x_1, x_2) = 0.74 \Rightarrow x_1 = f(x_2) \quad (7.12)$$

The function thus obtained can be substituted into the expression of the delta function (7.11). This provides us with the values of Δ for a fixed utility as a function of only one of the issues, and can be obtained for other utility values in a similar way.

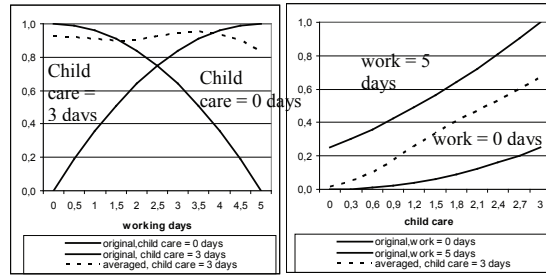


Figure 7.8: Original and averaged utility values running through maximum Δ -point.

The Δ -values obtained by weighted averaging with the polynomial weighting function and the Gaussian weighting function for utility equal to 0.74 are rather small for both (see Figure 7.7 (right)), but weighted averaging with a Gaussian function produces smaller approximation errors: it is almost twice as good. For bids with utilities of 0.9 the Δ -values (see Figure 7.7 (bottom)) rise in comparison with that of 0.7, however, the Gaussian weighting function still gives a better result. For bids with a utility of 0.5 (see Figure 7.7 (left)) the Δ -values are quite similar.

In figure 7.8, a worst case analysis is illustrated. It presents the utilities for extreme values of childcare (figure 7.8(left)) and for the number of working days (figure 7.8 (right)) that run through the maximum Δ -value, corresponding to the bid with 0 working days and 3 days of childcare. It shows that the evaluation function associated with 0 days of child care (0 working days) is almost mirrored with respect to the evaluation function associated with 3 days of child care (5 working days). In effect, this shows that our child care example presents a serious test for our WAID method that somehow has to average these differences.

7.4.5 Bid Search Algorithm

The negotiation algorithm that is used plays a key role in obtaining a good negotiation outcome. The approximation of a preference profile allows an agent to more efficiently compute counter bids during negotiation, but does not in itself provide a guarantee that against arbitrary opponents a good negotiation outcome will be reached. More insight is required to assess the effects of using approximations of real preference profiles.

A transformation of the utility space will have an effect on the negotiation process as well as on the negotiation outcome. To assess the impact of the WAID method, a negotiation strategy is applied to the employment contract example. Here, we use the ABMP-strategy proposed in [10].

The ABMP-strategy, outlined in Table 7.1, determines a bid in two steps: the strategy first (a) determines the target utility for the next bid, and then (b) determines a bid that has that target utility. The (b) part of the strategy is very efficient for independent utility spaces. For the purpose of comparison, however, we can use exhaustive search through

the complete utility space to find a bid in the second step, provided that the space is discretized in a suitable manner (using small enough steps). In this way, the first step (a) in the ABMP-strategy followed by the second step (b) using exhaustive search can be applied to the original utility space whereas the original ABMP strategy can be applied to its approximation.

An additional check is incorporated into the strategy when the approximated utility space is used to avoid the risk of accepting bids with low utilities in the original space that have much higher utilities in the approximated space. The bids with high Δ -values, that have shifted significantly due to application of the averaging method, can be filtered out in this additional step. When the agent receives a bid from its opponent, the agent has to calculate the associated original utility as well and compare it with the bid acceptance threshold.

We propose a parameterized procedure that can be used to control the probability of large outcome deviations. The parameters of this procedure can moreover be used to influence the tradeoff between the accuracy of the negotiation outcome and the computational efficiency of the negotiation strategy. In the next sections, experimental results are presented that allow the tuning of these parameters.

In the negotiation algorithm the bid selection procedure is the source of the deviation of the negotiation outcome. In particular, in step 3 of the algorithm described in the previous section the approximated space is used instead of the original space which gives rise to outcome deviations. To avoid approximation errors that are too big, we propose to add a checking procedure in this step which compares the utility of a bid in the approximated space with the utility in the original space.

Table 7.1: Negotiation algorithm with bid search procedure

- 1 **Evaluate bid $bid_A(i)$ received from opponent A:**
Accept and end negotiation if $u_b(bid_A(i)) > u_b(bid_B(i))$
- 2 **Compute concession and target utility:**
Concession $\gamma = \beta * (1 - \mu / u_B(bid_B(i))) * (u_B(bid_A(i)) - u_B(bid_B(i)))$
Target utility: $\tau = u_B(bid_B(i)) + \gamma$
- 3 **Determine a next bid:**
 - 3a **Find a bid with target utility**
Find a bid $bid_B(i+1)$ such that $u'_B(bid_B(i+1)) \approx \tau$
 - 3b **Compare bid utility in approximated and original space**
Check whether $|u_B(bid_B(i+1)) - u'_B(bid_B(i+1))| < \delta$
 - 3c **If not, find next candidate for the bid and repeat step (3b):**
Find next candidate bid $bid_B(i+1)$ such that $u'_B(bid_B(i+1)) \approx \tau$
and utility with previous bid only differs minimally
- 4 **Else, send bid to opponent**

The proposed procedure integrated into the negotiation algorithm can be found in Table 7.1. The step to determine a next bid is refined and an iterative procedure is incorporated

to check whether the difference in utility stays below a certain threshold δ . As before, in step 3a a bid is computed that matches a certain target utility. In step 3b, however, now a check has been incorporated that checks whether $\Delta U(bid) < \delta$, that is, whether the absolute approximation error stays below a threshold δ . This additional check itself is computationally cheap, since it involves only a simple calculation using equation (4). If $\Delta U(bid) > \delta$, a bid bid' , which utility differs minimally from the previously computed bid, is searched for, until $\Delta U(bid') < \delta$. This iterative procedure for finding an appropriate bid is called δ -checking.

The additional check is used to avoid the risk of proposing bids with (very) low utilities in the original space that have (much) higher utilities in the approximated space. The concessions made in step 3 thus are controlled by a parameter δ to ensure that they are not too big.

The δ -checking procedure introduces additional search again into the computation of a bid. Various heuristics could be applied again, however, to minimize the amount of search. For example, a limit on the number of iterations could be introduced for spaces of high dimensionality to ensure a bid would be found within a reasonable amount of time. (The probability of finding an appropriate bid is high in high-dimensional spaces close to the mpoint.) The relation of the value of the δ -parameter and the computational cost is analyzed in more detail using experimental results in Section 7.5.4.

7.5 Experimental Evaluation

In this section, we present experimental results that show how the value of the δ -parameter in the checking procedure relates to the distribution of the outcome deviation. These results show that there is a direct relation between the size of δ and outcome distribution.

7.5.1 Case Study

In this section, the negotiation strategy outlined in 7.1 is used to study the bids that an agent will offer during a negotiation using the original as well as the approximated utility space. The negotiation strategy that an agent decides to use should not only fit the agent's personality profile and culture, its experience in general and the current domain and negotiation partner, but it also has to be applicable given the utility space.

In our experiments, the same profile of the employer was used in the original as well as in the approximated case. The employer's profile that has been used is the same as that introduced above.

Figure 7.9 (left) shows the outcome space build up out of the utilities of the employer and employee per bid. Each point on the chart represents one bid. The coordinates of the

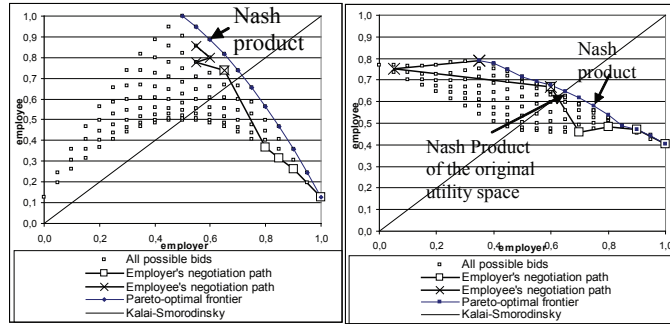


Figure 7.9: Outcome space, optimality criteria, and negotiation paths (left) for the original utility space of the employee, and (right) for the approximated utility space of the employee.

bid are the utilities of the opponents (x-coordinate is the employer's utility of the bid, y-coordinate is the employee's utility of the bid). The Nash product representing a bid with the highest utilities simultaneously for both opponents of the original utility space equals 0.53 and corresponds to a bid of 5 working days with 2.5 days of childcare, which satisfies the employee's constraints. The Kalai-Smorodinsky solution is 1.5 days of child care and 5 working days. This bid is found by locating a bid on the Pareto-optimal frontier, which is closest to the line drawn from points with utilities of (0; 0) to points with utilities (1; 1). This bid represents a negotiation outcome where both parties get the same utility. Using the ABMP strategy with exhaustive search for both parties, the negotiation lasts 4 rounds (4 bids from each side, the employer starts) and finishes when the employee accepts a bid of 2 days of childcare with 4.5 working days.

Figure 7.9(right) presents the result using the original ABMP strategy for both parties, where the profile of the employee has been approximated. The bids in the utility space are now concentrated around the employees original and approximated utility level of 0.7 (the *m*-point) with some spread towards lower utilities. The Nash product shifts to the bid of 5 working days and 1.5 days of childcare and the Kalai-Smorodinsky solution now is 4 working days and 1.5 days of childcare.

The original outcome space and the approximated one are significantly different. However, the difference is not critical for the negotiation itself due to the fact that most of the bids for which the difference is significant will not be used in a negotiation and we basically aim for the efficient solutions (Kalai-Somorindinsky point, and Nash Product). Also note that the bids are shifted only on the vertical axis (employee's utility), because the employer's profile remains the same. The negotiation performed for the same setup but using the approximated employee's utility space is also finished in 4 rounds as in the previous experiment and also results in a deal of 4.5 working days and 2 days of child-care. This example shows that the approximation procedure leads to some shifts in the efficient outcomes of the negotiation with respect to Nash and Kalai-Smorodinski. However, it also confirms that these bids and those around them preserve their meaning for the negotiator. Negotiation outcomes for both utility spaces are rather close even though the

negotiation paths are different.

7.5.2 Impact on Outcome Deviation

To analyze the impact of the WAID method on the negotiation outcome and computation costs a probabilistic experimental setup has been used. The negotiation outcomes obtained by using the WAID are compared with those obtained using the original utility space. The experimental results are obtained from utility spaces modeled by multivariate quadratic polynomials. Such polynomials are widely-used in decision making theory to represent preferences of a decision maker, e.g. see [15]. They are more expressive than the classic multilinear or multiplicative forms studied by Keeney and Raiffa in [11] and, thus, can cover wide range of domains including scheduling, assignment, quality control, facility layout, computer-aided process planning, and others (see [15]). On the other hand, efficient methods for preference elicitation using polynomial representation exist and studied in [15]. These methods are based on the fact that the polynomial can be easily differentiated and the decision maker can use gradients of a polynomial function to assess its utilities.

These polynomials may have multiplicative terms $x_i x_j$ which represent issues:

$$u(x_1, x_2, \dots, x_n) = \sum_l^n w_l \sum_{i=0}^n \sum_{j=0}^n a_{i,j} x_i x_j, \text{ where } x_0 = 1 \quad (7.13)$$

Values of the coefficients $a_{i,j}$ are generated randomly from interval $[-1; 1]$. Then, the evaluation functions are normalized to interval of $[0; 1]$ using a Monte-Carlo method. It is well-known that solving such quadratic programming problems is NP-hard, see e.g. [7]. The m -point parameter that has to be fixed in order to apply the WAID-method is determined for each utility space by a Monte-Carlo method.

The ABMP negotiation algorithm [10] (see Table 7.1) is used to assess the outcome deviation that may occur when an approximated space is used instead of the original space during a negotiation. In the experiments that were performed agent A also uses a variant of the ABMP strategy but does not approximate any issue dependencies in its utility space. Instead it uses exhaustive search through its utility space in step 3 to determine a next bid given a suitable discretization of this space (i.e. using small enough steps). To compare outcomes for utility spaces of medium size, the same negotiation is performed again with agent B using exhaustive search in step 3. Of course, exhaustive search can only be used for utility spaces of medium size due to exponential time costs and memory limitations. It is, however, imperative to use it if we want to calculate outcome deviation. In the experiments, spaces with up to a number of 5 issues and a number of discretization steps of at most 25 have been used (see also Section 7.5.2 and 7.5.3).

The main result of the experiments performed shows that the distribution of negotiation outcome deviations is similar to a normal distribution with a mean value close to zero.

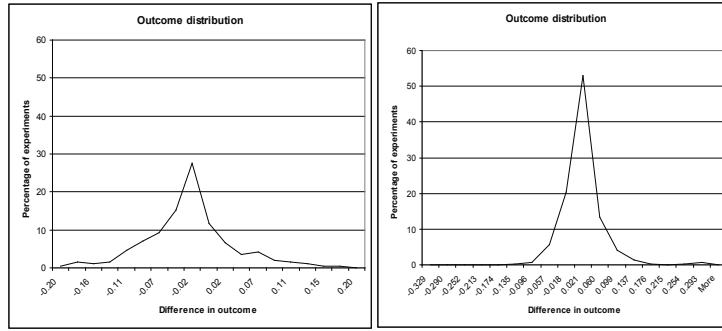


Figure 7.10: Distribution of negotiation outcome deviation without checking procedure (left) and with the procedure (right) for approximated spaces vs. original spaces for 4 issues ($k=15$).

Figure 7.10 (left) presents the distribution of outcome deviations for a negotiation about 4 issues. The deviation is a result of using the approximated space in the negotiation strategy instead of performing an exhaustive search to find a good bid in the original space. We use absolute values of the deviation in terms of utility instead of percentages here to have uniform presentation of the parameters and results throughout the paper. Note, that evaluation and utility functions are normalized into interval of $[0; 1]$. The bell-shaped distribution on the figure (average=-0.02; std.dev.= 0.09) means that the negotiation over the approximated space tends to produce the same result as the negotiation over the original space using exhaustive search. This demonstrates that one may expect to obtain reasonable outcomes when negotiating with approximated spaces instead of non-approximated spaces.

Even though this result shows that approximating the original utility space to remove issue dependencies may result in quite reasonable outcomes compared to those obtained otherwise, it also shows that there is quite a high chance of deviating significantly. In fact, for the 4 issue case figure 2 shows that there is a quite high probability of obtaining outcomes that are worse by up to 20%. Additionally, the curve is not really symmetrical and shows a tendency towards negative deviations. As an illustration, the probability of obtaining a result that is worse than 10% equals 0.196. It is clear that in many domains such a high risk will be unacceptable.

The impact of adding the δ -checking procedure to the negotiation algorithm on the outcome distribution is significant, as is shown by Figure 7.10 (right). The experimental setup is exactly the same but the negotiation algorithm used by agent B now includes the checking procedure. It shows the outcome distribution for a threshold of $\delta = 0.01$.

Clearly, the outcome distribution of the right plot in figure 7.10 is more symmetrical than in dashed curve and more clustered around the mean; it has a mean=-0.00016 and a standard deviation of 0.045. A more detailed analysis of the relation between Δ and the outcome deviation is presented in the Section 7.5.2.

The main conclusion thus is that additional measures need to be taken to reduce this risk.

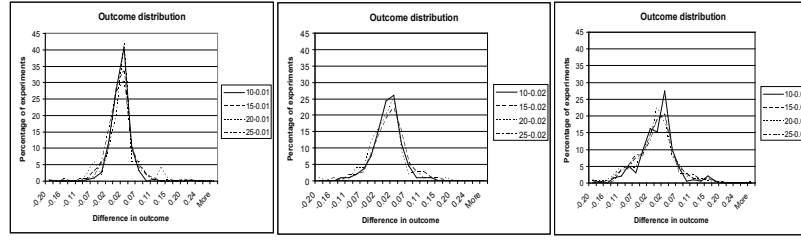


Figure 7.11: The distribution of outcome deviations for 5 issues and $\delta = 0.01$ (left), $\delta = 0.02$ (middle), $\delta = 0.03$ (right). The various lines relate to different k -values.

The benefit of using approximated spaces is clear: issues can be negotiated independently which makes the negotiation tractable. Controlled balance has to be found between the computational costs and the risk of significantly deviating negotiation outcomes.

Additionally, we investigated the influence of the discretization per issue under consideration on the outcome distribution. In the experiments we performed, the possible values for each issue were reduced by discretizing the space to 10, 15, 20, and 25 values. In order to assess the impact of adding the checking procedure to the negotiation algorithm, we performed experiments with 3, 4, 5, and 6 issues. Finally, for the δ -parameter of the checking procedure we used the values 0.001, 0.005, 0.01, 0.02, 0.03, and 0.05. In total, we performed over 44.000 experiments in which the outcomes were compared with the original space: 12.000 for 3 issues, 12.000 for 4 issues, 12.000 for 5 issues, and 6.000 for 6 issues. Comparisons of negotiation outcome for spaces of higher dimensionality were not feasible. The higher the number of issues n and the higher the discretization parameter k , the longer it takes to do the exhaustive search (it takes kn steps). To investigate the scalability of the proposed approach, we ran in total 500 experiments with 7,8,9,10 and 15 issues for $\delta=0.02$ and each k -value, so 2000 experiments in total. The results for 10 and 50 issues are presented in Section 7.5.4.

The experimental results relating the value of δ to the outcome distribution are depicted in Figure 7.11. We do not show all results but only those for δ -values of 0.01, 0.02, and 0.03 which most clearly demonstrate the impact of different values on the distribution and also define the turning points where decreasing this parameter further does not have a very big impact anymore (see also Figure 7.13) and decreasing it results in significantly worse outcomes.

In Figure 7.11, on the x-axis the outcome difference is set out. The outcome deviation may be bigger than the value of the δ -parameter since errors may accumulate over multiple rounds in the negotiation. The y-axis refers to the percentage of experiments having particular outcome differences. The different lines correspond with different values of the discretization parameter k . For each combination of a particular number of issues, δ -value, and k -value, 500 experiments were run.

In general, as is to be expected since δ is supposed to control the error introduced by the approximation, the experimental findings show that smaller values for δ result in

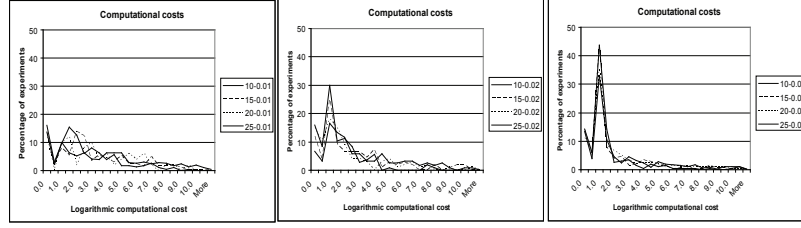


Figure 7.12: Computational costs for 5 issues and $\delta = 0.01$ (left), $\delta = 0.02$ (middle), $\delta = 0.03$ (right). The various lines relate to different k -values.

negotiation outcomes that are closer to the outcomes in the original space.

The findings illustrated in Figure 7.11 are as follows. For $\delta = 0.01$ (see Figure 7.11(left)) the standard deviation ranges from 0.0327 to 0.0442, and the average outcome difference ranges from -0.0066 to 0.0015. For $\delta = 0.02$ (see Figure Figure 7.11(middle)) the standard deviation ranges from 0.0350 to 0.05806 and the average outcome difference ranges from -0.0142 to 0.0010. Finally, for $\delta = 0.03$ (see Figure Figure 7.11(right)) the standard deviation ranges from 0.0499 to 0.0717, and the average outcome difference ranges from -0.0199 to -0.0151.

7.5.3 Impact on Computational Cost

Including the checking procedure implies that the bid determination part might need iterations to find an appropriate bid. The previous section shows that smaller δ -values lead to better outcome deviations, and it stands to reason that the smaller the value, the higher the number of iterations needed. To get more insights into the frequency with which the need for iterations causes high computational costs, a series of experiments have been performed. The algorithm was tested for 4, 5, 6, and 10 issues, with the discretization value k varying over $\{10, 15, 20, 25\}$ and δ varying over $\{0.005, 0.001, 0.03, 0.02, 0.01\}$. Each test was performed 500 times with randomly generated original utility spaces.

Figures 7.12 shows the results for 5 issues, the results for other values are not shown, since they do not provide additional insights. In these pictures, on the x-axis the logarithmic costs are set out. The y-axis refers to the frequency with which an experiment had such a logarithmic cost, with respect to the total number of experiments. The different lines refer to different k -values.

The results clearly show the expected increase of high computational costs for higher δ -values: higher percentages for higher computational values. However, when looking at the areas underneath the lines, another interesting observation can be made. In Figure 7.12(left), for $\delta = 0.01$, the bulk of the area underneath the lines ends approximately at $\ln(x) = 6$. In Figure 7.12(middle), for $\delta = 0.02$ the bulk ends at $\ln(x) = 4$, and in Figure 7.12(right), for $\delta = 0.03$ at $\ln(x) = 2$. Evidently, the number of iterations needed is bounded.

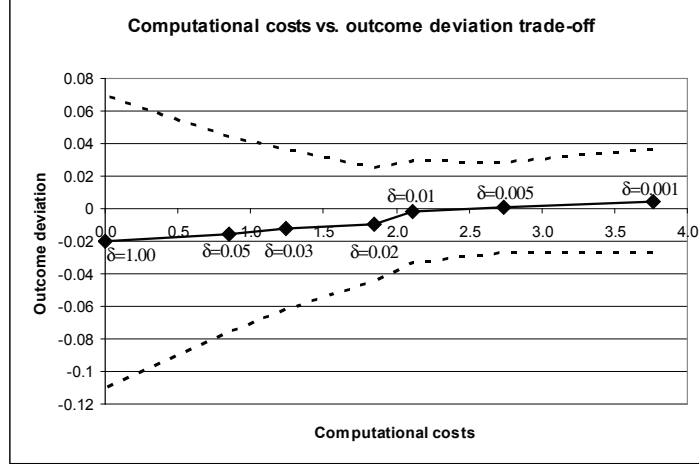


Figure 7.13: Computational cost vs. outcome deviation for 5 issues and $k=10$.

7.5.4 Tuning Computational Costs

Combining the results of the outcome analysis of Section 7.5.2 and the computational cost analysis of Section 7.5.3 shows that the need for a small outcome difference has to be balanced against computational costs. In this a setting for the k , and δ parameters is chosen that balances accuracy against efficiency.

To find a good balance between accuracy and cost, an integrated analysis has been performed for the usual combination of parameters: the number of issues ranging over $\{4, 5, 6, 10\}$, k ranging over $\{10, 15, 20, 25\}$ and δ ranging over $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.05, 1\}$. $\delta = 1$ corresponds to a setting without checking procedure.

Figure 7.13 presents the trade-off between negotiation outcome accuracy and the computational costs. Each point on the solid line of the chart represents the average of a series of experiments where δ varies over $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.05, 1\}$. The top dashed line is an average+std.dev. and bottom dashed line is the average-std.dev.

The results show that a good compromise is a δ -value of 0.02: for $\delta < 0.02$ the costs increase, for $\delta > 0.02$ the outcome approximation gets worse. Furthermore, the standard deviation drops off at this value, but does not decrease further for even smaller δ -values.

To analyse the scalability of the modified negotiation algorithm we performed a series of negotiations with 10-issues. Exhaustive search as a benchmark for the negotiation is no longer possible due to the extremely large utility space. Figure 7.14 shows average of the computational cost depending on the number of issues for $\delta = 0.02, k = 20$. The figure suggests that the most of the randomly generated utility spaces remain tractable for the negotiation algorithm with the δ -checking procedure.

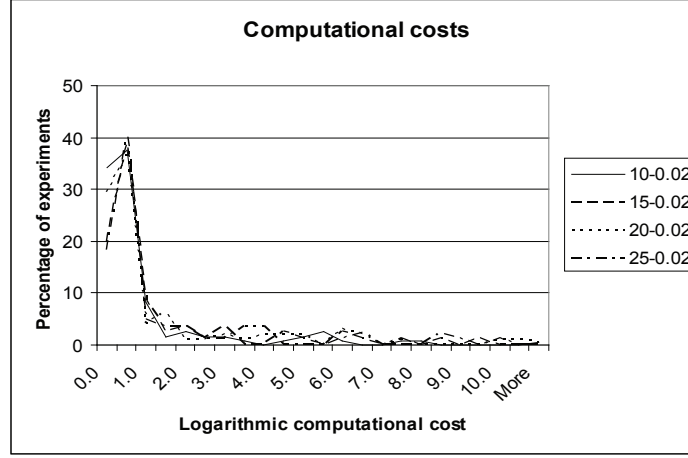


Figure 7.14: Computational costs for 10 issues and $\delta = 0.02$. The different lines refer to different k -values

7.6 Conclusions

In this paper we introduced a new approach that allows agents to deal with complex utility functions in a negotiation environment with interdependent issues. Instead of representing the negotiation task as an optimization task for interdependent issues we propose an approximation method to simplify the agent's utility using the observation that in common negotiation settings the expected negotiation outcome is approximately known and the insight that the nature of utility spaces for such common negotiation settings has enough structure to make our approach applicable. The method provides a means to analyze the impact of the approximation on a particular utility space, thereby making it possible to determine up front, whether or not the approximation is useful in any particular domain.

The main advantage of the proposed method is that it enables applicability of a wider range of computational negotiation strategies without introducing a mediator into the negotiation. Available information about the domain and the most probable negotiation outcome can be used to increase the accuracy of the method in the utility area around the expected outcome, which is most important for the negotiation.

However, using an approximation always comes with a risk. In the case of multi-issue negotiation, the risk is that a bid is proposed (and accepted by the other party) that seems to have a good utility, but in fact, in the original utility space has a much lower utility. The δ -checking procedure proposed in this paper offers a way to avoid this risk at the cost of additional computations. Experimental results show, however, that a tradeoff can be made between the accuracy of the bids and the computational overhead this entails. If the δ -parameter in the checking procedure is set to 0.02, the utility of the bids made is at most 0.02 away from the real utility, on a scale from 0 to 1. Moreover, using this value for the δ -parameter, the negotiation algorithm including the δ -checking procedure can handle high-dimensional utility spaces. the negotiation outcome obtained in this manner

only slightly deviates from the outcome obtained without approximation.

The additional check that compares the utility of exchanged bids with the utility of the original utility space during a negotiation prevents an agent from accepting low-utility bids in the original space with a high error in the approximated space. This check in itself is computationally cheap and ensures reasonable negotiation performance.

To conclude, in this paper an effective balance is found of accuracy versus efficiency for multi-issue negotiation with issue dependencies in which the dependencies are removed by approximation.

Bibliography

- [1] Y. Bar-Yam. *Dynamics of complex systems*. Addison-Wesley, 1997.
- [2] R. Davies and R.G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109, 1983.
- [3] James S. Dyer, Peter C. Fishburn, Ralph E. Steuer, Jyrki Wallenius, and Stanley Zionts. Multiple criteria decision making, multiattribute utility theory: The next ten years. *Management Science*, 38(5):645–654, May 1992.
- [4] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [5] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.
- [6] Shaheen Fatima, Michael Wooldridge, and Nick Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In Sierra Decker, Sichman and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, May 2009.
- [7] P. Favati, G. Lotti, and F. Romani. Theoretical and practical efficiency measures for symmetric interpolatory quadrature formulas. *BIT Numerical Mathematics*, 34(4), 1994.
- [8] Koen Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the AAMAS 2008*, 2008.
- [9] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces. In *Proc. of IJCAI-2007*, pages 1347–1352, 2007.
- [10] C.M. Jonker and J. Treur. An agent architecture for multi-attribute negotiation. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on AI, IJCAI'01*, pages 1195 – 1201, 2001.

- [11] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives*. Cambridge University Press, 1976.
- [12] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Paper 125 of the Center for eBusiness@MIT*. <http://ebusiness.mit.edu>, 2001.
- [13] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation Journal*, 12, November 2 2003.
- [14] Guoming Lai, Cuihong Li, Katia Sycara, and Joseph Andrew Giampapa. Literature review on multi-attribute negotiations. Technical Report CMU-RI-TR-04-66, Robotics Institute, Carnegie Mellon University, December 2004.
- [15] Behnam Malakooti and Sriram Subramanian. Generalized polynomial decomposable multiple attribute utility functions for ranking and rating multiple criteria discrete alternatives. *Applied Mathematics and Computation*, 106(1):69–102, December 1999.
- [16] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [17] D. G. Pruitt. *Negotiation Behavior*. Academic Press, 1981.
- [18] Howard Raiffa. *The Art and Science of Negotiation*. Belknap Press of Harvard University Press, Cambridge, Mass., 1982.
- [19] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2003.
- [20] V. Robu, D.J.A. Somefun, and J.A. La Poutre. Complex multi-issue negotiations using utility graphs. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, pages 280–287, Utrecht, The Netherlands, 2005.
- [21] J.S. Rosenschein and G. Zlotkin. *Rules of encounters Designing conventions for automated negotiation among computers*. Cambridge, MA: MIT Press, 1994.
- [22] C. Sierra, P. Faratin, and N.R. Jennings. A service-oriented negotiation model between autonomous agents. In M. Boman and W. van de Velde, editors, *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in Multi-Agent World, MAAMAS'97*, volume 1237 of *Lecture Notes in Artificial Intelligence*, pages 17–35. Springer-Verlag, 1997.
- [23] Koye Somefun, Tomas B. Klos, and Han La Poutre. Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles. In Marijn Janssen, Henk G. Sol, and René W. Wagenaar, editors, *Proceedings International Conference on Electronic Commerce 2004*, pages 361–370. ACM, 2004.
- [24] Leigh Thompson. *The Mind and Heart of the Negotiator*. Prentice Hall, 2004.

- [25] J. Wallenius, J. Dyer, P. Fishburn, R. Steuer, S. Zionts, and K. Deb. Multiple criteria decision making and multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science*, 54, 2008.

Chapter 8

Conclusions and Future Work

Negotiation is a research topic addressed from various disciplines: management studies, social science, and computer science/artificial intelligence. Negotiation is the most important process that allows people to form alliances, cooperations, and underlies all trade. In this thesis we address single-session negotiations from an artificial intelligence perspective. Single-session negotiations are negotiations in which people negotiate once on a particular topic. This subclass of negotiations occurs frequently in real-life.

This thesis proposes a principled design method for efficient and generic strategies for single-session negotiations. Our method is based on principles of heuristics and game theory, full analysis of outcome, dynamics, and domain, best-practice design, and empirical evaluation. In this way, our method takes into account the inherent complexity of the variation in negotiation domains (including scale and issue-dependencies), negotiators (human and software agents), and negotiation settings (number of parties involved and protocol).

These principles were derived by applying an analytical method proposed in Chapter 2 to state-of-the-art automated negotiating strategies. The method extends the conventional outcome analysis with an analysis of dynamic properties of a negotiation strategy. Fundamental is the use of a classification of negotiation moves and the introduction of negotiation behaviour metrics. We show how the analytical part of our design method was used to evaluate our new strategies in which the comparison to state-of-the-art negotiation strategies and the way humans negotiate plays an important role. Our hypothesis that the domain of negotiation might influence the performance of a strategy was confirmed. Our analysis of domains, strategies, and dynamics of negotiations led to a number of design guidelines for the design of efficient negotiation strategies:

- Knowledge about the opponent is essential to reach near Pareto-efficient outcomes.
- The strategy must be tested in domains with different characteristics, such as predictability of the issues (i.e., domain knowledge can be used to predict issue preferences), size of the domain, dependencies between issues.

- The strategy must be tested against a range of strategies, as well as against humans.
- The strategy must be tested for a range of profiles (for user and opponent).
- Study the dynamics of the strategy in tournaments using the properties formulated in the analytical method.
- Challenge other researches and/or students to outperform the strategy in a repeated negotiation setting, i.e., a setting where they can learn over multiple sessions.

To support our design method according to these guidelines, in Chapter 3 we developed the General Environment for Negotiation with Intelligent multi-purpose Usage Simulation (GENIUS). The main purpose of GENIUS is to evaluate a strategy in a simulated negotiation. For that purpose GENIUS offers a range of tools: an analytical toolbox, a tournament simulator, a preference elicitation user interface, user interfaces for human negotiators. It includes a repository with a number of negotiation domains and negotiating agents. To minimize the programming efforts, GENIUS provides an application programming interfaces (API) to access negotiation protocols, domains, preference profiles, and a skeleton of a negotiating agent. An existing negotiating strategy can be integrated in GENIUS by means of adapters which are used to wrap the code.

One of the main criteria formulated in Chapter 2 concerns the fact that an efficient strategy needs (at least to approximate) a model of the opponent's preferences. Therefore, in Chapter 4, a learning technique to learn opponent's preferences in single-session negotiations, called BOP, is proposed. Using the Bayesian learning BOP is able to construct a model of both the preferences associated with issue values and the weights that rank the importance of issues to an opponent. The learning algorithm allows for the incorporation of prior available opponent knowledge, e.g., derived from domain knowledge or previous interactions, to improve learning performance. However, it does not require any such knowledge. BOP was tested in a rigorous experimental setup using GENIUS. The learning performance of BOP was studied in various negotiation settings according to the proposed design method, see 4 for details. In addition, our results show that the efficiency of offers generated by a strategy can be improved by using a model of opponent's preferences learned by BOP.

In Chapter 5 we propose a behaviour-based negotiation strategy that uses an (real or learned) opponent model to classify the opponent's moves, select a response move, and search for efficient counteroffers. In every move the strategy tries to increase the chance of acceptance by moving towards the Pareto optimal frontier without conceding. The response move (counteroffer) is selected using the last opponent's move in a Tit-for-Tat manner. The strategy was instantiated with the BOP learning algorithm to construct a model of the opponent's preferences. The strategy was implemented and tested in GENIUS. Evaluation results of the strategy in GENIUS show a significant increase in utility of negotiation outcomes obtained by the proposed strategy compared to the state-of-the-art strategies.

In Chapter 6 we extend the results obtained in this thesis to one-to-many negotiations.

We used the BOP learning technique to approximate Qualitative Vickrey Auction (QVA) [4] with a negotiation setup. This auction is a generalization of the well-known Vickrey auction to a general complex multi-issue setting where payments are not essential. The QVA requires the buyer (a single negotiating party on one side) to publicly announce its preferences to all sellers (multiple negotiating parties on the other side). Given public preferences it can be shown that the outcome is Pareto efficient and the mechanism is strategy-proof.

To overcome the limitation of public preferences we propose a negotiation protocol that consists of multiple negotiation rounds in which buyer does not have to announce its preferences publicly. Instead, the sellers observe the buyer's bidding behaviour and try to learn buyer's preference profile using the BOP learning technique. Our experimental results show that the negotiation protocol approximates the outcome of QVA. Furthermore, we show in numerous experiments that the BOP learning technique in combination with a concession strategy dominates a Zero Intelligence strategy [3] with random offers.

Finally, in Chapter 7 we addressed the problem of issue dependencies in negotiation domains. Such issue dependencies result in non-linear utility spaces. Finding good bids in such spaces is computationally complex as the space grows exponentially in the number of issues. We propose an off-line method that approximates the original non-linear utility space with a linear space. The method minimizes the distance between the original space and its linear approximation, especially in the area of the potential agreement. Note, that such an approximation comes with the risk that a bid is proposed (and accepted by the other party) that seems to have a good utility, but in fact, in the original utility space has a much lower utility. We propose a checking procedure that offers a way to avoid this risk at the cost of additional computations. The experimental results show that a balance between risk and additional computations can be determined per negotiation domain.

Future Work

We believe that our results show the need for benchmark negotiation problems. An interesting direction for future research in this area would be to propose measures for exploitability and robustness of a negotiation strategy. A good negotiation strategy must be able to withstand a seemingly weird opponent strategy, such as Random Walker, as well as strategies that try to exploit it.

In real-life negotiations domains are not fixed; new issues can be discovered during the whole negotiation process. Therefore, future work is to design an adaptable version of BOP and GENIUS (including the analytical framework) that can handle a change in the number or structure of issues.

Furthermore, we plan to integrate the BOP learning technique and the NMS strategy in the Pocket Negotiator a negotiation support system that assists human negotiators. The strategy as well as the learning technique can be used for various purposes. For example,

a model of the opponent preferences learned by BOP can be presented to the user thus enabling the user to improve the efficiency of his/her offers. The NMS strategy can be used to give an advice about the next negotiation move. The form of the advice requires further investigations.

The BOP algorithm can be initialized with a priori knowledge about probable preferences in a specific negotiation domain. Such knowledge can be learned from previously stored negotiations, if available. Clustering techniques can be applied to a set of opponent models (learned by BOP or gathered in a different way) to find typical opponent profiles for a specific domain, and recurring themes over domains. The Pocket Negotiator can store such opponent profiles in a repository that can be used in future in case of a recurrent negotiation with the same opponent or with a different opponent in the same domain. Such information can be shared between users.

An important direction for future work is negotiation in domains with issues dependencies. In chapter 7 we show that interdependencies increase computational complexity of a negotiation strategy. Most of the existing negotiating strategies are designed for linear additive utility functions. In this thesis we propose an approximation method to allow such strategies to deal with negotiation domains with issue dependencies. The method decreases computational complexity of a search in a utility space for a bid with a given utility. The learning algorithm proposed in this thesis is based on an assumption that the opponent's preferences can be modelled without issue dependencies. In future, we will study influence of issue dependencies in a negotiation domain on the learning performance. Furthermore, we will extend the learning algorithm to deal with issue dependencies that cannot be tolerated.

Recently, qualitative models for preferences have received a significant attention in the negotiation [5] and decision making research [1, 2]. This work is motivated by the fact that in some negotiation domains it is difficult to elicit quantitative models (e.g., based on utility functions) from humans, since humans generally express their preferences in a more qualitative way. Thus, in our future work we plan to extend and modify the proposed techniques for the qualitative preference representation models.

Heuristic methods used in this thesis focus on the exchange of proposals that denote single points in the negotiation space without any additional information. The only feedback that can be received from an opponent is a counter-proposal, which itself is another point in the space. The argumentation-based approach typically extends classic negotiation protocols with a possibility to exchange arguments. This information explains explicitly the opinion of the agent making the argument. In future, we will investigate when and what arguments a negotiating agent should use in a negotiation to improve the negotiation efficiency.

Bibliography

- [1] Craig Boutilier. Toward a logic for qualitative decision theory. In *In Proceedings of the KR'94*, pages 75–86. Morgan Kaufmann, 1992.

- [2] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [3] Dhananjay K. Gode and Shyam Sunder. Allocative efficiency in markets with zero intelligence (zi) traders: Market as a partial substitute for individual rationality. *In Journal of Political Economy*, 100(1):119–137, 1993.
- [4] P. Harrenstein, M.M. de Weerd, and V. Conitzer. A Qualitative Vickrey Auction. *In Proceedings of the EC*. ACM Press, 2009.
- [5] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Argumentation-based preference modelling with incomplete information. *In Proceedings of 10th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA-X)*, 2009.

Samenvatting

Het hoofddoel van dit proefschrift is het ontwikkelen van generieke en efficiënte geautomatiseerde strategieën voor bilaterale onderhandelingen waarin de twee onderhandelende partijen hun voorkeuren niet expliciet kenbaar maken. Een onderhandelingsstrategie is het beslissingsmechanisme dat bepaalt welke acties de onderhandelaar dient te nemen. De term generiek verwijst in dit proefschrift naar het idee dat een dergelijke strategie niet op voorhand hoeft te beschikken over kennis van het onderhandelingsdomein of de tegenpartij. Een strategie dient dus generiek te zijn in die zin dat deze succesvol kan worden toegepast in elk willekeurig onderhandelingsdomein en dat deze afgestemd kan worden op domein specifieke kenmerken voor het behalen van nog betere onderhandelingsresultaten. De term efficiëntie verwijst in dit proefschrift naar het feit dat een strategie in staat moet zijn om op effectieve wijze te onderhandelen met zowel geautomatiseerde agenten als menselijke onderhandelaars zodat het uiteindelijke onderhandelingsresultaat voor geen van beide partijen voor verbetering vatbaar is. Het ontwerp van de in dit proefschrift voorgestelde onderhandelingsstrategie is gebaseerd op analyses van *state-of-the-art* onderhandelingsstrategieën. Daarbij is gebruik gemaakt van een eveneens in dit proefschrift ontwikkelde analytische methode. Doordat deze methode de dynamische eigenschappen van een onderhandelingsstrategie analyseert, vormt deze methode een significante bijdrage aan de huidige onderhandelingsijkpunten (*negotiation benchmarks*). Een van de belangrijkste bevindingen van dit deel van het onderzoek is dat een strategie de voorkeuren van de tegenpartij in een onderhandeling dient te leren om de efficiëntie van deze onderhandelingen te verhogen. Deze bevindingen zijn in lijn met de bevindingen uit de in de management wetenschappen en sociale wetenschappen beschikbare literatuur over onderhandelen. Wij hebben onze resultaten in het leren van de voorkeuren van de tegenpartij in onderhandelingen toegepast in een *one-to-many* onderhandelingsituatie. Tevens hebben wij het probleem van afhankelijkheden tussen verschillende onderhandelingskwesties (*negotiation issues*) geadresseerd. De afhankelijkheden tussen onderhandelingskwesties vormen een onoverbrugbare hindernis in het huidige onderzoek naar onderhandelingsstrategieën. Wij hebben daarvoor een benaderingsmethode ontwikkeld die deze afhankelijkheden elimineert. Hoewel op het eerste gezicht dit deel van het onderzoek een zijspoor lijkt te vormen in dit proefschrift, was het niettemin van fundamenteel belang dat wij dit probleem hebben aangepakt om zo de schaalbaarheid en toepasbaarheid van onze onderzoeksresultaten aan te tonen.

Kort samengevat zijn de onderzoeksvragen waarop dit proefschrift is gebaseerd de vol-

gende:

1. Hoe kunnen wij *state-of-the-art* geautomatiseerde strategieën ontwerpen voor *multi-issue* bilaterale onderhandelingen waarin enkel biedingen worden uitgewisseld?
2. Wat voor een analytisch raamwerk is essentieel voor het ontwikkelen van dergelijke geautomatiseerde strategieën?
3. Is het mogelijk om in een onderhandeling het preferentieprofiel van de tegenpartij te leren gegeven het feit dat enkel een sequentie van biedingen wordt uitgewisseld?
4. Kunnen wij deze preferentieprofielen op een effectieve wijze gebruiken in geautomatiseerde biedingsstrategieën?
5. Kunnen wij onze resultaten toepassen in *one-to-many* onderhandelingsituaties?
6. Kunnen wij de onderhandelingsruimten van onderhandelingen met afhankelijkheden tussen de onderhandelingskwesaties benaderen door gebruik te maken van onderhandelingsruimten zonder dergelijke afhankelijkheden?

Onderhandelen is een type interactie, waarbij twee of meerdere op het eigen gewin gerichte agenten (elk met eigen doelstellingen, behoeften en gezichtspunten) raakvlakken proberen te ontdekken inzake onderhandelingskwesaties en proberen een akkoord te bereiken om een kwestie van wederzijds belang te regelen of om een conflict op te lossen (cf. [1]). Mensen onderhandelen zowel in hun persoonlijke leven als in hun professionele leven [2]. Alhoewel de meeste mensen zich zelf beschouwen als effectieve onderhandelaars, blijkt uit tal van experimenten dat zij niet het onderste uit de kan halen (*“leave money on the table”*) (cf. [2]). Vooruitgang op het gebied van kunstmatige intelligentie [3] en managementwetenschappen [2] bood inspiratie voor de ontwikkeling van de eerste ondersteuningssystemen voor mensen tijdens onderhandelingen, alsmede voor de ontwikkeling van software-agenten die uit hun naam onderhandelen.

Dit proefschrift concentreert zich op generieke en efficiënte biedingsstrategieën voor eenmalige onderhandelingsessies tussen twee onderhandelaars. De biedingsstrategieën kunnen worden ingezet door onderhandelende software-agenten. Voor de nadruk op eenmalige onderhandelingsessies is gekozen omdat diverse belangrijke onderhandelingen in het echte leven eenmalige onderhandelingsessies zijn. Voorbeelden hiervan zijn het kopen van een huis, het kopen van een auto of het voeren van arbeidsonderhandelingen. In technisch opzicht brengt deze beperking tot eenmalige onderhandelingsessies met zich mee dat het niet mogelijk is te leren van voorgaande ervaringen met een bepaalde onderhandelingspartner¹.

In dit proefschrift betogen wij dat de ontwikkeling van generieke en efficiënte biedingsstrategieën een analytisch raamwerk behoeft voor grondige evaluatie van biedingsstrategieën.

¹Ter verduidelijking noemen wij de twee onderhandelende partijen in een bilaterale onderhandeling de gebruiker en de tegenpartij. De software-agenten die wij ontwikkelen handelen altijd in het belang van de gebruiker. De andere onderhandelende partij wordt de tegenpartij genoemd. Hierbij hebben wij volledig kennis genomen van de argumenten die door de Harvard Business School zijn aangevoerd om de term tegenpartij te vermijden.

Hiervoor hebben wij de General Environment for Negotiation with Intelligent multi-purpose Usage Simulation (GENIUS) ontwikkeld. Wij tonen aan dat een juiste analyse van onderhandelingsstrategieën het bestuderen van de dynamiek van de onderhandeling omvat en niet louter het bestuderen van de onderhandelingsresultaten. Dit laatste is typisch voor het huidige *state-of-the-art* onderzoek naar geautomatiseerde onderhandelingen. Hiertoe hebben wij een reeks aan dynamische eigenschappen ontwikkeld die hun nut in onze analyses hebben bewezen. Deze eigenschappen zijn opgenomen in de analytisch omgeving die onderdeel uitmaakt van het GENIUS raamwerk.

Onze analyse van het *state-of-the-art* onderzoek naar geautomatiseerde bilaterale onderhandelingsstrategieën hebben de volgende belangrijke criteria opgeleverd voor het ontwikkelen van generieke en efficiënte biedingsstrategieën:

- Kennis over de tegenpartij is essentieel om resultaten te bereiken, die de toestand van Pareto-efficiëntie dicht naderen.
- De strategie moet worden getest in domeinen met verschillende karakteristieken. Hiervoor hebben we domein karakteristieken ontwikkeld zoals voorspelbaarheid, grootte en afhankelijkheden. Tevens, hebben wij in GENIUS een *repository* van domeinen opgenomen.
- De strategie moet worden getest met een reeks van profielen (zowel voor de gebruiker als voor de tegenpartij). Daarom hebben wij in GENIUS voor elk domein een *repository* van profielen opgenomen.

Aangezien veel *state-of-the-art* onderhandelingsstrategieën niet om kunnen gaan met afhankelijkheden tussen onderhandelingskwesties, hebben wij een benaderingsmethode voor het elimineren van deze afhankelijkheden ontwikkeld. Deze methode kan worden gecombineerd met die strategieën die, voordat ze de tegenstander een bod te doen, naar onderhandelingsbiedingen zoeken met een bepaalde utiliteit.

Onze analyses hebben aangetoond dat *state-of-the-art* onderhandelingsstrategieën in essentie op concessie gebaseerde strategieën zijn. Kenmerkend voor deze strategieën is dat ze op de tegenbiedingen van de tegenpartij geen reactie geven waaruit de aanvaardbaarheid van die tegenbieding zou kunnen blijken. Deze strategieën garanderen niet altijd dat er pas een concessie wordt gedaan als de tegenpartij een soortgelijke concessie doet. Bovendien richten de ontwikkelaars van *state-of-the-art* onderhandelingsstrategieën zich niet op het feit dat, om een aanvaardbare overeenkomst te bereiken in de onderhandelingen, de kans op een overeenkomst zou moeten worden gemaximaliseerd. In dit proefschrift hebben wij een generieke en op gedrag gebaseerde strategie ontwikkeld en getest, die uitdrukkelijk rekening houdt met deze knelpunten. Deze strategie heet de Nice Matching Strategy (NMS), daar zij de gebruik maakt van een *tit-for-tat* aanpak om zo een goede uitkomst voor de agent zelf veilig te stellen en daar zij zogenaamde vriendelijke biedingen doet om zo de kans dat de tegenpartij een bod accepteert te maximaliseren. De NMS strategie maakt gebruik van een op Bayesiaans leren gebaseerde techniek om de voorkeuren van de tegenpartij te leren: Bayesian learning algorithm for Opponent Preferences (BOP). NMS gebruikt dit geconstrueerde model van de tegenpartij om een soort

spiegel-strategie te implementeren. Het spiegelen is een geavanceerde variant van de *tit-for-tat* strategie. Onze analyse toont aan dat NSM superieur is aan de *state-of-the-art*.

Het leren van de voorkeuren van de tegenpartij tijdens een onderhandeling, dat wil zeggen zolang er biedingen worden uitgewisseld, is essentieel in de context van eenmalige onderhandelings sessies. Dit is een uitkomst van onze analyse methode voor onderhandelingsstrategieën. Onze analyse toont tevens aan dat zonder kennis van het model van de tegenpartij generieke biedingsstrategieën niet efficiënt zijn. Dit leerdoel is met name een uitdaging daar waar we aannemen dat onderhandelingen gesloten zijn, dat wil zeggen, daar waar de enige beschikbare informatie bestaat uit de biedingen die worden uitgewisseld. BOP is geëvalueerd in zowel een bilaterale onderhandelings situatie als in een *one-to-many* onderhandelings situatie. Het BOP leermechanisme dat we hebben ontwikkeld is geëvalueerd met behulp van de GENIUS omgeving waaruit bleek dat BOP de voorkeuren van de tegenpartij tijdens eenmalige onderhandelings sessies goed kan benaderen.

De methode die wij hebben gebruikt om generieke en efficiënte biedingsstrategieën te ontwikkelen is gebaseerd op een iteratieve aanpak waarin GENIUS een centrale rol speelt :

1. Allereerst, door het toepassen van GENIUS hebben wij de sterke en zwakke punten (inefficiënties) van bestaande biedingsstrategieën geïdentificeerd.
2. Ten tweede, van de analytische resultaten aldus verkregen is gebruik gemaakt om nieuwe technieken te identificeren die de efficiëntie verhogen en om verbeterde strategieën te ontwikkelen.
3. Ten derde, met behulp van de ontwikkelingskit van GENIUS zijn de voorgestelde technieken geïmplementeerd.
4. Ten vierde, de validiteit van de nieuwe strategieën is getest en geanalyseerd door gebruik te maken van GENIUS om tegen de strategieën te onderhandelen in de diverse onderhandelings domeinen verzameld in de GENIUS *repositories* door het houden van een GENIUS toernooi.

De methode die wij hebben gebruikt om generische en efficiënte biedingsstrategieën te ontwikkelen is gebaseerd op een herhaalde benadering waarin GENIUS een centrale rol speelt:

1. Eerst, door GENIUS toe te passen identificeren wij sterke punten en zwakheden (ondoelmatigheden) van bestaande biedingsstrategieën
2. Ten tweede, worden de zo verkregen resultaten gebruikt om nieuwe technieken te identificeren die de efficiëntie verhogen en zo betere strategieën te ontwikkelen
3. Ten derde, wordt de ontwikkelingsuitrusting van GENIUS gebruikt om de voorgestelde technieken te implementeren
4. Ten vierde, wordt de validiteit van de nieuwe strategieën getest en geanalyseerd door het houden van toernooien tegen de verschillende strategieën en in diverse

onderhandelingsdomeinen die in de repository van GENIUS worden verzameld.

Wij sluiten deze samenvatting af met een lijst van de belangrijkste kenmerken van GENIUS. De GENIUS omgeving is ontwikkeld tijdens dit promotie traject als een volwaardig hulpmiddel voor onderzoek. De belangrijkste kenmerken zijn:

- Het ondersteunt de implementatie en toetsing van nieuwe strategieën.
- Het bevat *repositories* bestaande uit onderhandelingsdomeinen, preferentieprofielen per domein en strategieën die de analyse van onderhandelingsagenten faciliteren voor een reeks aan verschillende opstellingen.
- Het verschaft een grafische gebruikersinterface voor het construeren en toevoegen van nieuwe onderhandelingsdomeinen en preferentieprofielen.
- Het verschaft een grafische gebruikersinterface waarmee mensen tegen bekende of onbekende tegenstanders kunnen onderhandelen. De tegenstanders kunnen mensen of software-agenten zijn.
- Het verschaft een toernooi-omgeving die de gebruiker in staat stelt om diverse strategieën tegen elkaar te laten onderhandelen in elke combinatie van onderhandelingsdomeinen en preferentieprofielen.
- Het verschaft een analytische omgeving die de onderzoeker ondersteunt in het analyseren van de gegevens die uit het *runnen* van een toernooi worden verkregen.

Bibliography

- [1] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2003.
- [2] L.L. Thompson. *The Heart and Mind of the Negotiator*. Pearson Prentice Hall, 2005.
- [3] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley Sons, second edition, 2009.

Acknowledgements

I would like to start with expressing my gratitude to my promotor Catholijn Jonker and copromotor Koen Hindriks who guided me through the process of preparing this thesis. Catholijn became my supervisor when I started working on my master thesis in Amsterdam. She taught me how to bring my ideas to the attention of others. Then Koen joined our small team when we moved to Nijmegen. Ever since he was a daily supervisor for me. I learned from him not only how to set goals and make plans but also how to achieve them. Thanks to you I also learned to look critically at own results. Dear Koen and Catholijn, it has been my pleasure working with your! I would also like to thank the reading committee for their feedback.

I would like to continue my “thank you” list with my VICI-colleagues Tim Baarslag, Willem Paul Brinkman, Joost Broekens, Iris van de Kieft, Alina Pommeranz, Birna van Riemsdijk, Wietske Visser, and Pascal Wiggers. I enjoyed working with. I would like to thank my colleagues from the Man-Machine Interaction with whom I had a pleasure to work. I would like to thank Duco Ferro for the nice discussions we had during coffee breaks and for helping me as a paranimf. Special thanks for Wouter Pasma who had been my room-mate at TU Delft for several years. I would like to thank Zhenke Yang for sharing his experiences with arranging the defense.

I want to thank the “simulants” from the TACT team: Gert-Jan Hofstede, Sebastiaan Meijer and Tim Verwaart. Thanks to you now I can simulate a lot of different things in various ways. My special thanks to Tim Verwaart for giving me opportunity to do an internship at LEI and for willingness of being my paranimf.

I would like to thank my family. First of all, my father Vyacheslav Tykhonov who introduced the world of science to me. I would like to thank my mother Larysa Tykhonova for taking my passions with patience. I would like to thank my grandparents who also made their important contribution into raising me as a researcher. And of course I want to thank my wife Sofiya for her patience and support. I want to thanks Irina Polonska and Jacques Fraissard who became a part of my family. I would like to thank Valentina Sokolova and Jacob de Raat who have been my second family here The Netherlands.

Last but not least I would like to thank my friends for their support. I want to thank Borys Omelayenko for wonderful discussion we had about doing research. I would like to thank Joost Schildwacht for showing me a universe outside the academia. I would like also mention Vera Kartseva, Maksym Korotkiy, Zhanna Serdyukova. If the reader could not

find his or her name in these acknowledgements then it means that I want to say the word of gratitude to you personally.

List of SIKS Dissertations - SIKS Dissertatiereeks

====
1998
====

- 1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects
- 1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information
- 1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective
- 1998-4 Dennis Breuker (UM)
Memory versus Search in Games
- 1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

====
1999
====

- 1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling;
Automated modelling of Quality Change of Agricultural Products
- 1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets
- 1999-3 Don Beal (UM)
The Nature of Minimax Search
- 1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects
- 1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven
Specification of Network Information Systems
- 1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems
- 1999-7 David Spelt (UT)
Verification support for object database design
- 1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent
Mechanism for Discrete Reallocation.

====
2000
====

- 2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance
- 2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

- 2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie;
een procesbenadering en actorperspectief.
- 2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design
- 2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.
- 2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication
- 2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management
- 2000-8 Veerle Coup (EUR)
Sensitivity Analysis of Decision-Theoretic Networks
- 2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization
- 2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations,
Algorithms and Architecture
- 2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

- 2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks
- 2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models
- 2001-3 Maarten van Someren (UvA)
Learning as problem solving
- 2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with
Instance-Based Boundary Sets
- 2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style
- 2001-6 Martijn van Welie (VU)
Task-based User Interface Design
- 2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization
- 2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.
- 2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes
- 2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice

BRAHMS: a multiagent modeling and simulation language
for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

====
2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments
inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative
E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

=====
2003
=====

- 2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments
- 2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems
- 2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
- 2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology
- 2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach
- 2003-06 Boris van Schooten (UT)
Development and specification of virtual environments
- 2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks
- 2003-08 Yongping Ran (UM)
Repair Based Scheduling
- 2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour
- 2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction
between medium, innovation context and culture
- 2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
- 2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval
- 2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models
- 2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
- 2003-15 Mathijs de Weerd (TUD)
Plan Merging in Multi-Agent Systems
- 2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to
Digital Media Warehouses
- 2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing
- 2003-18 Levente Kocsis (UM)
Learning Search Decisions

=====
2004
=====

- 2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic
- 2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business
- 2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
- 2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures
- 2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity
- 2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques
- 2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes
- 2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politieke gegevensuitwisseling en digitale expertise
- 2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning
- 2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects
- 2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies
- 2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents
- 2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play
- 2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium
- 2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining
- 2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning
- 2004-17 Mark Winands (UM)
Informed Search in Complex Games
- 2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models
- 2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval
- 2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

====
2005

=====

- 2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications
- 2005-02 Erik van der Werf (UM))
AI techniques for the game of Go
- 2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language
- 2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data
- 2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing
- 2005-06 Pieter Spronck (UM)
Adaptive Game AI
- 2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems
- 2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications
- 2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages
- 2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
- 2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
- 2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry
- 2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen
- 2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics
- 2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes
- 2005-16 Joris Graaumans (UU)
Usability of XML Query Languages
- 2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components
- 2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks
- 2005-19 Michel van Dartel (UM)
Situated Representation
- 2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives
- 2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by
Exploiting Application Semantics

=====
2006
=====

- 2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting
- 2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations
- 2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems
- 2006-04 Marta Sabou (VU)
Building Web Service Ontologies
- 2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines
- 2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools
for Graphical Service Modeling
- 2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering
- 2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web
- 2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion
- 2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems
- 2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types
- 2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts
- 2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents
- 2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
- 2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain
- 2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks
- 2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device
- 2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing
- 2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach
- 2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

- 2006-21 Bas van Gils (RUN)
Aptness on the Web
- 2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation
- 2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web
- 2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources
- 2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC
- 2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval
- 2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories
- 2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

=====
2007
=====

- 2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures
- 2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach
- 2007-03 Peter Mika (VU)
Social Networks and the Semantic Web
- 2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach
- 2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance
- 2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs
- 2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings
- 2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations
- 2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation
- 2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
- 2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System
- 2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty
- 2007-13 Rutger Rienks (UT)

Meetings in Smart Environments; Implications of Progressing Technology

- 2007-14 Niek Bergboer (UM)
Context-Based Image Analysis
- 2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model
- 2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems
- 2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice
- 2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations
- 2007-19 David Levy (UM)
Intimate relationships with artificial partners
- 2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network
- 2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005
- 2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns
- 2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems
- 2007-24 Georgina Ramírez Camps (CWI)
Structural Features in XML Retrieval
- 2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

====
2008
====

- 2008-01 Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach
- 2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations
- 2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach
- 2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration
- 2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
- 2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
- 2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning
- 2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference

- 2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective
- 2008-10 Wauter Bosma (UT)
Discourse oriented summarization
- 2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach
- 2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation
- 2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks
- 2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort
- 2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.
- 2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective
- 2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises
- 2008-18 Guido de Croon (UM)
Adaptive Active Vision
- 2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
- 2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten
- 2008-21 Krisztian Balog (UVA)
People Search in the Enterprise
- 2008-22 Henk Koning (UU)
Communication of IT-Architecture
- 2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia
- 2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching
- 2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency
- 2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
- 2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design
- 2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks
- 2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans
- 2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
- 2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure

====
2009
====

2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU)
Understanding Classification

2009-07 Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law and the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
Operating Guidelines for Services

2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT)

- Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System
- 2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets
- 2009-20 Bob van der Vecht (UU)
Adjustable Autonomy: Controlling Influences on Decision Making
- 2009-21 Stijn Vanderlooy (UM)
Ranking and Reliable Classification
- 2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence
- 2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment
- 2009-24 Annerieke Heuvelink (VUA)
Cognitive Models for Training Simulations
- 2009-25 Alex van Ballegooij (CWI)
"RAM: Array Database Management through Relational Mapping"
- 2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services
- 2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web
- 2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models
- 2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications
- 2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage
- 2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text
- 2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors
- 2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?
- 2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach
- 2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
- 2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks
- 2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks
- 2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
- 2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets

- 2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language
- 2009-41 Igor Bereznyy (UvT)
Digital Analysis of Paintings
- 2009-42 Toine Bogers
Recommender Systems for Social Bookmarking
- 2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
- 2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations
- 2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful
- 2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion

====
2009
====

- 2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter
- 2010-02 Ingo Wassink (UT)
Work flows in Life Science
- 2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents
- 2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments
- 2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems
- 2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI
- 2010-07 Wim Fikkert (UT)
A Gesture interaction at a Distance
- 2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments
- 2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging
- 2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children
- 2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning
- 2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis
- 2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques
- 2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration

- 2010-15 Lianne Bodestaff (UT)
Managing Dependency Relations in Inter-Organizational Models
- 2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice
- 2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
- 2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation
- 2010-19 Henriette Cramer (UvA)
People's Responses to Autonomous and Adaptive Systems
- 2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
- 2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation
- 2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to
Heterogeneous Linked Data
- 2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions
- 2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies
- 2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective
- 2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines