

Complex practical negotiations based on autonomous agents

Citation for published version (APA):

Chen, S. (2014). *Complex practical negotiations based on autonomous agents*. [Doctoral Thesis, Maastricht University]. Maastricht University. <https://doi.org/10.26481/dis.20141021sc>

Document status and date:

Published: 01/01/2014

DOI:

[10.26481/dis.20141021sc](https://doi.org/10.26481/dis.20141021sc)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

COMPLEX PRACTICAL NEGOTIATIONS BASED ON
AUTONOMOUS AGENTS

COMPLEX PRACTICAL NEGOTIATIONS BASED ON
AUTONOMOUS AGENTS

DISSERTATION

to obtain the degree of Doctor at
Maastricht University,
on the authority of the Rector Magnificus,
Prof. dr. L. L. G. Soete,
in accordance with the decision
of the Board of Deans,
to be defended in public
on Tuesday October 21, 2014 at 16:00 hours

by

Siqi Chen

Supervisor:

Prof. dr. G. Weiss

Co-supervisor:

Prof. dr. K. P. Tuyls (University of Liverpool)

Assessment Committee:

Prof. dr. ir. R. L. M. Peeters (*chairman*)

Prof. dr. ir. J. C. Scholtes

Prof. dr. P. McBurney (King's College London)

Prof. dr. HF. Leung (The Chinese University of Hong Kong)

Dr. M.H.M. Winands

Printed by Optima Graphic Communication, Rotterdam, The Netherlands.
ISBN 978-94-6169-582-6

© Siqu Chen, 2014.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically, photocopying, recording or otherwise, without prior permission of the author.

To my beloved parents.

Acknowledgements

It is incredibly difficult to thank all of the people that have helped me, in one way or another, over the past three years. Firstly, I would like to thank both of my supervisors Prof. Gerhard Weiss and Prof. Karl Tuyls for all their time, friendship and intellectual inspirations during my Ph.D study at the Department of Knowledge Engineering (DKE), Maastricht University. This superb supervisory team has provided me with excellent guidance as well as energy and faith to complete this work and a host of workshop, conference and journal publications with respect to it.

I would also like to thank all members of SwarmLab. It was my fortune to work with so many great people there (Daan Bloembergen, Sjriek Alers, Daniel Claes, Josha Fossel, Haitham Bou Ammar, Dr. Frans Oliehoek and Dr. Kurt Driessens), your kindness and expertise in multi-agent systems proved invaluable. A special thank should go to Bijan Ranjbar-Sahraei and Frederik Schadd – you make my life in Maastricht really different and enjoyable! Besides, I want to thank my colleagues Firat Ismailoglu, Nasser Davarzani, Li You, Zhenglong Sun and Wei Zhao for their friendship and uncountable funny talks.

Furthermore I would like to acknowledge Prof. Ho-fung Leung for his supervision during my visit at the Chinese University Of Hong Kong. I also thank my friend Dr. Jianye Hao's help and many stimulating discussions. I also appreciate the help and support from the Interactive Intelligence Group at Delft University of Technology (specifically, Prof. Catholijn Jonker, Dr. Reyhan Aydogan, Dr. Tim Baarslag). Much of my research was based on the negotiation platform developed by the group.

I would like to express my gratitude for all the anonymous referees of my publications, those unknown people that gave me so many good questions and interesting communications on the occasion of a series of AI conferences and forums (MATES' 2012, ECAI'2012 & 2014, AAMAS'2012 & 2013 & 2014, IJCAI'2013, etc).

Maybe the most important part of my life is my mother who played above all a vital role in finishing this dissertation. I really thank her love and encouragement, both before and during my Ph.D studies.

Siqi Chen
September, 2014

Contents

Acknowledgements \diamond vii

Contents \diamond ix

Part I: Introductory Part \diamond 1

1 Introduction \diamond 3

1.1 Problem Definition \diamond 3

1.1.1 **Subproblem I:** Design of Effective Strategies for Agents . . . 5

1.1.2 **Subproblem II:** Game-theoretic Analysis of Performance . . . 6

1.2 Structure of the Thesis \diamond 7

2 Background \diamond 9

2.1 Automated Negotiations \diamond 9

2.1.1 Negotiation Forms 10

2.1.2 Negotiation Protocol 12

2.1.3 Concession Strategies 13

2.2 Complex Practical Negotiation \diamond 15

2.2.1 Zero Prior Opponent Knowledge 15

2.2.2 Continuous-time Constraints 15

2.2.3 Discounting Effect and Reservation Value 16

2.2.4 Negotiation model 16

2.3 State of the Art \diamond 17

2.3.1 Heuristic Approaches 17

2.3.2 Game Theoretic Approaches 20

2.3.3 Argumentation-based Approaches 21

2.3.4 Automated Negotiating Agent Competition 22

Part II: Design of Automated Negotiation Strategies \diamond 23

3 Strategies based on Decomposition Techniques \diamond 25

3.1 OMAC \diamond 25

3.1.1 Discrete Wavelet Transformation	26
3.1.2 Opponent Modeling	29
3.1.3 Decision-making Mechanism	32
3.1.4 Experimental Analysis	34

3.2 EMAR \diamond 38

3.2.1 Opponent Modeling	38
3.2.2 Adaptive Decision-making	43
3.2.3 Experimental Analysis	47

4 Strategies based on Gaussian Processes \diamond 53

4.1 OMAC* \diamond 54

4.1.1 Gaussian Processes	54
4.1.2 Opponent Modeling	56
4.1.3 Adaptive Concession-Making Mechanism	58
4.1.4 Counter-Offer Response Mechanism	61
4.1.5 Empirical Evaluations	62

4.2 Dragon \diamond 72

4.2.1 Sparse Pseudo-input Gaussian Processes	73
4.2.2 Opponent-modeling Component	76
4.2.3 Concession-making Component	78
4.2.4 Responding Component	80
4.2.5 Empirical Evaluations	81

Part III: Game-theoretic Analysis of Negotiation Strategies \diamond 89

5 Empirical Game-theoretical Analysis \diamond 91

5.1 Methodology \diamond 92

5.2 Underlying Two-Player Negotiation Games \diamond 93

5.3 Small Negotiation Tournament Games \diamond 95

5.4 Large Negotiation Tournament Games \diamond 96

5.5 Discussion \diamond 98

- 6 Evolutionary Game-theoretical Analysis \diamond 101
 - 6.1 Methodology \diamond 102
 - 6.2 Dynamics of Three Strategies with Local Interaction \diamond 103
 - 6.3 Dynamics of All Strategies with Local Interaction \diamond 107
 - 6.4 Discussion \diamond 109

Part IV: Concluding Part \diamond 111

- 7 Contributions and Future Work \diamond 113
 - 7.1 Contributions to the Research Questions \diamond 113
 - 7.2 Directions for Future Work \diamond 115

Bibliography \diamond 119

List of Figures \diamond 127

List of Tables \diamond 129

Addendum: Valorization \diamond 131

Summary \diamond 137

Samenvatting \diamond 139

Curriculum Vitae \diamond 143

Publication List \diamond 145

Part I

Introductory Part

1

Introduction

Negotiation is a process through which two or more parties that have conflictive interests try to reach a mutual agreement [55]. This process, seen as a coordination mechanism, is common in our everyday life in a variety of social, economic and political fields. In many cases negotiation is a time consuming process, and, moreover, finding good agreements can be a tough challenge for human negotiators, especially if they have no or just little negotiation experience or if the available negotiation time is limited [46]. Because of these limitations, there exists considerable interest in automating the negotiation process by means of software agents. In this context, a software agent (or an agent for short) is a piece of software that acts on behalf of a human negotiator and is able to interact autonomously with other negotiating agents or humans in order to reach agreement on a range of issues that are of conflicting interest to the involved parties [84]. The negotiations that are operated by autonomous agents are so-called automated negotiations. In contrast with human-based negotiations, automated negotiations can take very complicated forms because agents have less restrictive computational power and memory than humans [81]. Recent years have witnessed a rapidly growing interest in automated negotiations, mainly due to the broad application range in domains as diverse as electronic commerce and electronic markets [45, 63], supply chain management [78], task and service allocation [19], and combinatorial optimization [22].

In the remainder of this chapter we first describe the addressed problem and research questions and then outline the overall structure of this thesis.

1.1 Problem Definition

In automated negotiation, which nowadays is one of the core topics in agreement technologies and artificial intelligence [51, 72], autonomous agents are engaged in an automatic offer exchange process that gradually leads to a final solution in form of a mutual agreement. In general, automated negotiation is considered as a means to

solve conflicts and to achieve coordination among agents [39, 50]. More specifically, in automated negotiation computational autonomous agents attempt to arrive at joint agreements in competitive consumer-provider or buyer-seller scenarios on behalf of humans [39]. Toward this end, automated negotiation requires from these agents to have a high level of decision autonomy, so that they can decide on their own and in real-time what offers and counter-offers to make in order to reach a satisfactory agreement. This objective is, however, difficult to achieve in practice, particularly for the case of complex negotiations.

The basic features of the complex negotiations considered in this thesis are as follows. First, the negotiating agents do not know each other (i.e., they have not encountered before) and thus have no information about the preferences or strategies of their respective opponents. This makes it difficult to efficiently reach an agreement, especially because none of the agents knows what offers the respective other agents may consider as attractive. Second, the negotiation is being executed with deadline and with discount. More precisely, this means that the negotiation runs under real-time constraints and the agents thus should take into consideration at each time point the remaining negotiation time. The final utility decreases over time according to a discounting factor, which implies that unnecessary delays should be avoided as they result in negative effects on the negotiation outcome. And third, computational efficiency is important because agents may have very limited computing resources. Negotiation scenarios showing these characteristics are particularly challenging but common in real-world applications. For example, in the case of open electronic sales platforms an agent may be engaged in bilateral multi-issue negotiations with a service provider agent which it has never met before, and if the negotiation agent runs on a small mobile device then computational efficiency can be particularly crucial. Moreover, it can be advantageous to reach an agreement with the provider agent as quickly as possible, especially if other agents are also interested in the same service or if a fast agreement results in a reduced price or a cost-free service extension. A negotiation setting showing the above features, often referred to as complex practical negotiation (further details on this are provided in Section 2.2), obviously places high demands on the negotiation abilities of the agents.

The goal to provide a feasible solution to practical complex negotiations leads to the following central problem statement:

Problem Statement: *How to design effective strategies for complex negotiations, and how to evaluate them effectively?*

This problem statement induces two subproblems to be addressed: first, how to develop effective strategies, and second, what are appropriate performance criteria of evaluating negotiation strategies?

1.1.1 Subproblem I: Design of Effective Strategies for Negotiating Agents

Research Question 1: *How to learn opponent models in complex negotiations?*

In automated negotiations, effectively learning an opponent model is crucial for a successful negotiation strategy [6, 31, 88]. Modeling the opponent's behavior is, however, practically challenging because negotiators usually do not reveal their true preferences and/or negotiation strategies in order to avoid that others exploit this information to their advantage [18, 66]. Current approaches tend to employ computationally extensive algorithms for precise modeling and/or make simplifying assumptions about the negotiation setting (i.e., the negotiation domain, the opponent's behavior, or both). As a consequence, the value of these approaches is rather restricted when applied to complex negotiations. There are, for example, many approaches that only deal with single-issue negotiation and others that rely on the assumption the opponents have a rather simple and thus an easily predictable behavior. In other words, these approaches do suffer, to different degrees, from oversimplification and/or computational complexity.

Furthermore, apart from the ability to learning an opponent model, the use of an adaptive concession strategy is a decisive factor for successful negotiation. This implies the following important question:

Research Question 2: *How to concede appropriately in environments of high uncertainty and dynamics?*

Due to the challenges imposed by complex negotiations, current concession strategies (e.g., the prominent tit-for-tat strategy) do not fulfill the requirements of complex negotiations. For example, in such types of negotiations the negotiators are facing the problems of setting up contracts over a number of issues instead of a single issue. When making concession in multiple-issue scenarios, the loss of one agent's utility may not be equal to the profit increase of the other agent, and this makes these scenarios significantly different from single-issue negotiations. In particular, this leads to inefficiency of the class of behavior-dependent strategies such as tit-for-tat. Time-dependent strategies, which are another type of classic concession approaches to automated negotiations, are also not very effective because they only take into account the remaining negotiation time but do neglect other important factors such as discounting factors and reservation values.¹

¹These factors are formally introduced in 2.2.4 (page 16).

1.1.2 Subproblem II: Game-theoretic Analysis of Negotiating Agents' Performance

The evaluation of an agent-based negotiation strategy requires to apply performance criteria. Currently the most common, standard criterion is the performance level achieved in a simulated tournament. While this can give a good indication of the performance in certain negotiation settings, a closer look shows that tournament-based evaluation has serious limitations. This leads to following two research questions.

Research Question 3: *In addition to measuring the performance of negotiation strategies in tournament-based (thus fixed) negotiation scenarios, how to analyze the performance of negotiating agents in dynamic scenarios?*

Available work [13, 14, 28, 82] on automated negotiation primarily investigates the performance of negotiation strategies from a mean-scoring perspective based on tournament settings (where each player runs, or stands for, a specific strategy). A shortcoming of this perspective is that it does not capture the performance of a negotiation strategy in dynamic settings in which the agents are allowed to change their strategies during negotiation in order to increase their profits.

Another common assumption implicitly made in performance analysis is that each agent interacts with (negotiates against) *all* other participating agents. While this assumption is acceptable for scenarios in which the number of negotiators is relatively small, it is unreasonable for scenarios where many agents are present and the interaction range of the individual agents is limited (i.e., the agents interact only locally rather than globally). Such practically relevant scenarios are neither captured by tournament-based analysis nor by EGT (Empirical Game Theory [41]) analysis, and this implies the fourth Research Question addressed in this thesis:

Research Question 4: *How to analyze the performance of a negotiation strategy in scenarios where a large population of negotiating autonomous agents is involved and the interaction range of the agents may be restricted?*

In many realistic multi-agent systems (just as in natural societies) an agent does not interact with all others. Locality is thus an important factor in all negotiations where “neighborhood” needs to be taken into consideration. However, negotiation performance with respect to locality in the presence of many opponents has not been well studied so far and so it is still unclear how this issue can be tackled in a satisfactory way.

1.2 Structure of the Thesis

The remainder of the thesis is structured as follows. Chapter 2 provides the reader with background knowledge and concepts needed to understand the subsequent chapters. The Research Questions 1 and 2 are addressed in Chapters 3 and 4. Chapter 3 proposes two schemes for decomposition-based negotiation strategies, called OMAC and EMAR. Chapter 4 presents two more advanced frameworks, called Dragon and OMAC*, which employ Gaussian processes to better learn opponent models and to better adopt to opponent behaviors.

Next, Research Questions 3 and 4 are addressed in Chapters 5 and 6. Chapter 5 analyzes, by means of empirical game theory, the performance of strategies in dynamic negotiation settings, where global interaction and a limited number of participants is assumed. Chapter 6 expands the number of potential participants and also considers the spatial structure of interaction, and exploits evolutionary game theory for performance analysis.

Finally, Chapter 7 summarizes the contributions to the four research questions and identifies promising directions for future work.

2

Background

This chapter provides relevant general background knowledge.¹ First, in Section 2.1, important concepts of automated negotiations are introduced. Next, in Section 2.2 characteristics of complex practical negotiations are described. Finally, in Section 2.3 related work is discussed, including heuristic approaches, game-theoretic approaches, and argumentation-based approaches to automated negotiation.

2.1 Automated Negotiations

Because of the broad spectrum of potential applications in industrial and commercial domains [3, 19, 61, 63, 78], automated negotiations gain steadily growing attention as a fundamental and powerful mechanism for managing interaction among computational agents. Generally speaking, two different research trends can be distinguished in automated negotiation models: game-theoretic and heuristic models. Game-theoretic models consider a negotiation to be a game played by two or more players with a certain set of actions. The main focus is on reaching optimal solutions under the assumptions of full rationality and (partial) availability of information regarding the strategies and preferences of the negotiating opponents. Such kind of models inherently, when applied to negotiation settings, may suffer from some assumptions that they are dependent on. For instance, full rationality implies that each agent always behaves optimally within the frame of its knowledge. In practice, however, this would require that an agent possesses unbounded computational resources to guarantee. Besides, the agents may not know and use the same set of strategies (instead, they may be heterogeneous in this respect). Moreover, to identify which strategy is applied by an opponent or the likelihood of its preferences over issues under negotiation is also a difficult task. This is because an agent may not have been involved in interactions with his negotiating opponents before and because the opponents are

¹Additionally, more detailed information about chapter-specific aspects of negotiation theory, game theory and machine learning is given in the individual chapters.

likely to be not willing to reveal their private information in order to avoid possible exploitation. Therefore, it is obvious that some key assumptions of game-theoretic models are very critical with respect to real-world negotiation scenarios.

By contrast, heuristic models aim at searching for solutions that are as close as possible to the optimum, instead of reaching optimum agreements. Thus, heuristic models make relatively weak assumptions compared to game-theoretic models; to be exact they just assume incomplete knowledge regarding the opponents and the environment, and provide computationally affordable solutions in order to achieve sufficiently good results. The work reported in this thesis belongs to this category of models, because it concerns negotiation with bounded computational resources and opponents whose behavior is unknown. In the following, main aspects of automated negotiation relevant to heuristic models are overviewed.

2.1.1 Negotiation Forms

Due to the diversity of applications of negotiations, automated negotiations comes in a number of forms [49, 50]. Below the most prominent forms relevant to this thesis are considered.

Single-issue versus Multi-issue Negotiations

In single-issue negotiations, only one issue, for example, price, quality or delivery time, is considered to be negotiated between parties. This can reduce the negotiation complexity to a large extent, also simplifying the learning process of opponent models. However, this on the other hand results in seriously limiting its application scenarios as in real world people tend to be concerned with negotiations with more than one issues.

Automated negotiations with multiple issues see more application potentials, but places a higher demand on the agents' capabilities of choosing proper negotiation moves. In such sort of negotiations, it is likely that there exist a number of proposals (offers) that are at the same utility level for an agent but significantly different for this agent's negotiating opponent. The complexity of negotiations (in terms of their outcome space) increases rapidly, however, as the number of considered issues grows. The agents, therefore, should be able to identify proposals that are mutually beneficial for both parties for increasing the quality and efficiency of the negotiation.

Bilateral versus Multi-lateral Negotiations

Bilateral negotiations refer to a negotiation setting in which two conflictive parties negotiate with each other to search for a mutually acceptable agreement. The structure

of this form of negotiations is relatively simple in that it requires only a basic one-to-one communication protocol. By contrast with bilateral negotiations, multi-lateral negotiations solve conflicts among a number of parties. The efficiency of multi-lateral negotiations tends to be relatively low, since the final solution has to be agreed upon by all participants, at the expense of huge communication and coordination efforts. A real-world example of multi-lateral negotiation over China's participation in the World Trade Organization, which took decades to come to an agreement. As a matter of fact, multi-lateral negotiations are far more complex than bilateral ones, due to the complicated communication mechanisms that may be needed, the possibly very large variety of different interests of the negotiators, the possibly huge number of variables and constraints involved in the negotiation process, and so forth. Currently only little work is available on automated multi-lateral negotiations, and most research efforts – like those described in this thesis – focus on bilateral negotiation settings.

Sequential versus Concurrent Negotiations

When it comes to a situation where one party needs to negotiate with multiple parties about the same item or service, two options are typically available: running sequential or concurrent negotiations [62]. For the former type, a negotiator sequentially conducts negotiations with other agents (e.g., in the case of client-provider model, a client talks with one service provider at a time), whereas for the latter type, a negotiator simultaneously deals with more than one other agents (e.g., a client talks with all available providers of a service). Concurrency offers several potential advantages for a negotiating agent. In particular, it allows an agent to achieve better learning results (based on a broader learning input through multiple interactions) and to make decisions on offers and counter-offers in a very flexible way (e.g., by using different negotiation tactics in concurrent negotiation tasks and/or by changing the tactics in a specified task dependent on progress made in other tasks). These advantages, on the other hand, come at the cost of additional managerial efforts and computation time for coordinating multiple concurrent tasks – and this gets even worse in many-to-many concurrent negotiations (i.e., every agent is doing multiple negotiations with others). Sequential negotiations are easier to study and analyze, both empirically and mathematically; and their profound understanding is an important step toward the realization of efficient concurrent negotiations. The work described here deals with sequentially negotiating agents.

Complete versus Incomplete Information

Usually an agent does not have sufficient relevant information about its opponents. Specifically, an opponent's utility function, negotiation strategy and reservation util-

ity are often unknown, because revealing this information would enable an agent to exploit the behavior of its opponent. If the negotiating agents would obtain complete information, they could easily determine the optimal moves among possible actions. In a negotiation with incomplete information, one agent is unlikely to know exactly how its actions affect the opponent and how an opponent reacts on offers proposed by it, and vice versa. As an effect, the ability to learn about opponents (their utility function, preferences, future offers, etc.) becomes important in such settings.

To summarize, in this thesis we concentrate on *Automated Bilateral Sequential Multi-issue Incomplete-information Negotiation* (e.g., [5, 12, 13, 17, 44, 82]), which is also common for many real-world negotiation scenarios. Characteristic to this type of negotiation is that there are two autonomous agents, each having only incomplete information about the negotiation behavior of the respective other agent, that exchange offers in turn with the goal to agree on a common set of issues such as price, delivery time, quantity and quality, where these issues may be of conflictive importance for the negotiators.

2.1.2 Negotiation Protocol

The protocol defines the basic rules regarding the negotiation, which describes the possible types of participants, the set of negotiation states, the set of actions that the participants can take in each state [39]. In simple terms, the negotiation protocol governs who is involved in the negotiation as well as how the negotiation should proceed. Some good examples of them are the simultaneous-offers and alternating-offers protocols.

Simultaneous Offers

Under the simultaneous-offers protocol, the involved parties should propose their offers at the same time. The Nash demand game [57] is an example of a simultaneous game, where the two parties simultaneously make a single offer. Consider an example of this game in which two persons negotiate over the sharing of one cheesecake. Players p and q make a single offer each, $O_p \in [0,1]$ and $O_q \in [0,1]$, which indicates how much of the cheesecake they wish to take. If the total amount of cake the players specified in their offers does not exceed one (i.e., $O_p + O_q \leq 1$), then the negotiation ends up with an agreement between these two players. Otherwise, if $O_p + O_q > 1$, no agreement can be reached, and the negotiation ends in conflict (or, negotiation failure). However, a reached agreement is likely not to be efficient, when some of the cake remains unallocated (i.e., $O_p + O_q < 1$). These inefficient outcomes that are not

Pareto-optimal², and thus fail to maximize the joint expectation of the participating parties.

Alternating Offers

According to the alternating offers protocol, the parties should in turn make offers and counter-offers. This process continues until one of the parties accepts the opponent's offer or, alternatively, one of the parties chooses to terminate the negotiation (without agreement), or the negotiation deadline is reached. Compared to the simultaneous offers protocol, the alternating offers protocol is more suitable for automated negotiation settings where the negotiators are usually given incomplete information. This is because, under a simultaneous protocol, it would become difficult to enforce that the offers of the two parties are compatible and Pareto-efficient without an effective coordination mechanism. To avoid further complexity, this thesis deals with automated strategies on the basis of the alternating-offers protocol.

2.1.3 Concession Strategies

The classic view of artificial intelligence with respect to negotiation in incomplete-information settings is that agents have to eventually concede so that an agreement could be settled [23, 24, 59]. Concession-making may take place in various magnitudes and at different stages of the negotiation process. Concession strategies, responding to counter-offers from opponents, determine the way of how and when the agents should compromise so that an agreement can finally be made. One of the most influential publications regarding concession strategies is arguably [23], in which several concession strategies are proposed that take into account various aspects of a negotiation; besides, there also exists another type of well-known concession strategies (e.g., ABMP) that are based on the difference of utilities of bids. These strategies are briefly characterized next.

Time-dependent Strategies

Time-dependent strategies consider how much remaining time is left in an ongoing negotiation as the most vital factor in concession-making. Specifically, among them three types can be further distinguished, namely, linear tactics, concenter tactics and Boulware tactics [23]. Linear tactics suggest the same amount of concession at each time point during negotiation until its required utility is as low as the given reservation utility. Against that, concenter tactics rapidly concede during the early phase of the

²An outcome is considered to be Pareto-optimal, if there are no other outcomes that a single participant could select with no other participants objecting [83].

negotiation process, whereas Boulware tactics concede very slowly in the beginning but turn to a friendly attitude when the negotiation process approaches the deadline.

Behavior-dependent Strategies

In the case of behavior-dependent strategies, the concession made by an agent is equivalent (or, more generally, proportional) to its benefit from the negotiation movements made by the opponent over the previous rounds. A famous representative of this family of strategies is tit-for-tat, which aims at mimicking the opponent concession. There exist a number of variants of tit-for-tat, such as random absolute tit-for-tat, which makes the absolute concession carried out by the opponent in the last offer with a small random deviation, and averaged tit-for-tat, which makes the average concession made by the opponent in a certain time window.

Resource-dependent Strategies

This family of strategies computes concessions by considering the pressure imposed by scarceness of a type of resource under negotiation. It can be regarded as a generalization of time-dependent strategies for which the remaining time is treated as a type of resources. The basic idea behind resource-dependent strategies is that the scarcer the resource is, the more eager should the agent be to obtain this resource. Many factors of negotiations can be taken into account resources such as remaining time, discounting factors, a negotiator's impatience, etc.

Agent Based Market Place Strategies

Another interesting concession model for bilateral negotiations is the Agent Based Market Place (ABMP) proposed by Jonker et al. [40]. ABMP suggests negotiation moves by comparing the utility difference of bids. The amount of concession is regulated by a number of parameters, for example, a concession factor that determines to what degree an agent is willing to making concessions, the negotiation speed (i.e., the extent of concessions the agent typically makes per negotiation round), the acceptable utility gap (i.e., the maximal difference between the target utility and the utility of an offer that is acceptable), and impatience factor (i.e., the inclination to break-off if the opponent's bids are too far away from the expected). Additionally, ABMP includes other interesting characteristics such as the possibility of sharing preference information with the other party, guessing heuristics that allows agents to determine the ranking of issues, and issue values based on the bid history.

2.2 Complex Practical Negotiation

Our work deals with a more interesting negotiation form – complex negotiations, in contrast to simple negotiations (no discounting effect, prior opponent information, etc). Complex negotiations allow a negotiation setting with much more freedom, e.g., negotiating agents may be given only a strictly-limited amount of time to consider his next move. They are therefore of relevance to a wider range of practical applications, and are also common to many human-human negotiation scenarios.

2.2.1 Zero Prior Opponent Knowledge

Information that would help an agent in calculating advantageous offers and reaching a satisfying agreement include the opponent's utility function, negotiation strategy and time constraints imposed on the negotiation process. In fact, in a negotiation in which each agent has complete information about the opponent, it is possible for each agent to calculate which proposals satisfy both sides best. Due to the conflicting nature of negotiations among self-interested negotiators, however, an opponent cannot be expected to be willing to reveal this information. Facing a negotiation with zero prior opponent knowledge, an agent cannot know exactly which proposals result in a high level of satisfaction and how the opponent will react to its proposals. What is thus required is that an agent is able to acquire useful opponent information in real-time during the negotiation process.

2.2.2 Continuous-time Constraints

A hard time constraint is usually supposed in form of a deadline by which a negotiation is to be completed. Continuous-time constraints define constraints as the limited amount of physically elapsed time, whereas discrete-time constraints specify the number of allowed interactions (or negotiation rounds). This means that in the case of continuous time constraints the negotiators are facing a real-time deadline by which an agreement has to be reached because otherwise the negotiation ends with a failure (this deadline is denoted by t_{max}).

Continuous-time constraints require from an agent to adapt its own negotiation behavior to the opponent strategy [81]. An agent should not concede too quickly to its opponent, unless the opponent also shows a positive attitude in choosing his actions. On the other hand, it is important that an agent does not follow a too tough approach (such as the Boulware strategy [24]) because this delays the negotiation unnecessarily, causing the value of an eventual agreement to be decreased. What the best choice is, however, depends on the opponent and the impact of its strategy on the available negotiation time. In such settings the likelihood that the agents fail to reach

an agreement gets increasing as the negotiation process proceeds and the remaining negotiation time decreases. Continuous-time constraints make it more difficult to predict the utility of an offer that an agent might make in the future moves. This is primarily because the time at which any future offer will be made by an agent depends on the time cost of its opponent generating counter-offers. For the same reason, it is also impossible to in advance know how many interactions will be left before the deadline is reached.

2.2.3 Discounting Effect and Reservation Value

Discounting effect is used as an indicator of the cost of the duration of a negotiation, with the purpose of encouraging agents complete a negotiation as soon as possible. It has the effect that any particular agreement that is reached earlier is of higher value than an agreement reached at a later time. This results in that the action of delaying an agreement (due to incapability or intentionally for the purpose of exploiting opponents) will be punished, as the value of the agreement will be reduced in proportion to the elapsed time [1, 81].

Reservation utility gives the minimum utility level that an agent can expect, in other words, an agent can obtain the same amount of utility indicated by the reservation utility even if a negotiation fails. Also it determines the maximum concession that can be made by an agent when playing against a tough opponent that does not want to concede. This maximum concession restricts the amount of experience available for learning about an opponent, which in turn puts further demands on the agents' learning abilities.

2.2.4 Negotiation model

We now formalize the model related to complex negotiations. Let $I = \{a, b\}$ be a pair of negotiating agents, i represent a specific agent ($i \in I$), J be the set of issues under negotiation, and j be a particular issue ($j \in \{1, \dots, n\}$ where n is the number of issues). The goal of a and b is to establish a contract for a product or service. Thereby a contract consists of a package of issues such as price, quality and quantity. Each agent has a minimum payoff as the outcome of a negotiation; this is called the reservation value ϑ . Further, w_j^i ($j \in \{1, \dots, n\}$) denotes the weighting preference which agent i assigns to issue j . The issue weights of an agent i are normalized, summing to one (i.e., $\sum_{j=1}^n (w_j^i) = 1$). During a negotiation session agent a and b act in conflictive roles that are specified by their preference profiles. In order to reach an agreement they exchange offers (i.e., O) in each round to express their demands. An offer is thereby a vector of values, with one value for each issue. The utility of an

offer for agent i is obtained by the utility function defined as:

$$U^i(O) = \sum_{j=1}^n (w_j^i \cdot V_j^i(v_{jk})) \quad (2.1)$$

where v_{jk} is the k -th value of the issue j and V_j^i is the evaluation function for agent i , mapping a value of issue j (e.g., v_{jk}) to a real number.

If no agreement is reached at the end or one side breaks off before reaching the deadline, the negotiation then ends up with the disagreement solution (and each agent obtains a reservation utility ϑ as its payoff). Note that the number of remaining rounds are not known and the outcome of a negotiation depends crucially on the time sensitivity of the agents' negotiation strategies. This holds, in particular, for discounting domains, in which the utility is discounted with time. We define a so-called discounting factor δ ($\delta \in [0, 1]$) and use this factor to calculate the discounted utility as follows:

$$D^\delta(U, t) = U \cdot \delta^t \quad (2.2)$$

where U is the (original) utility and t is the standardized time (i.e., $t \in [0, 1]$). As an effect, the longer it takes for agents to come to an agreement the lower is the utility they can achieve. Note that a decrease in δ increases the discounting effect.

Upon receiving a counter-offer from the opponent, O_{opp} , an agent decides on acceptance, rejection and withdrawal according to the interpretation of its reasoning model.³ For instance, the acceptance decision can be made in dependence on a certain threshold $Thres^i$: agent i accepts if $U^i(O_{opp}) \geq Thres^i$, and rejects otherwise. As another example, the decision could be based on differences in successive utilities.

2.3 State of the Art

Negotiation strategies can be broadly classified into game theoretic, heuristic and argumentation based approaches [39]. These classes are overviewed below and characterized with respect to the complex-negotiation criteria described in Section 2.2.

2.3.1 Heuristic Approaches

Negotiation has traditionally been investigated in game theory [59, 66], and over previous years it has also developed into a core topic in the field of multi-agent systems

³If the agents know each other's utility functions, they can compute the Pareto-optimal contract [66]. However, a negotiator will not make this information available to its opponent in competitive settings.

[50, 54, 79]. A number of approaches have been proposed that, much like the approaches described in this thesis, explore the idea to equip an agent with the ability to build a heuristic model of its opponent and to use this model for optimizing its negotiation behavior (see [29] for a good overview). Modeling the opponent's behavior, however, is practically challenging because negotiators usually do not reveal their true preferences and/or negotiation strategies in order to avoid that others exploit this information to their advantage [18, 66].

Current approaches tend to employ computationally extensive methods for precise modeling and/or make simplifying assumptions about the overall negotiation setting (i.e., the negotiation environment and the opponent behavior). For example, there are approaches that deal with single-issue negotiation and others that assume that the opponents have a rather simple (e.g., non-adaptive) behavior. In the following, representative model-based negotiation approaches are overviewed.⁴

Preference learning

Many of the available approaches aim at learning the opponents' preferences or their reservation values. Faratin et al. [24] propose a trade-off strategy to increase the chance of getting own proposals accepted without decreasing the own profit. The strategy applies the concept of fuzzy similarity to approximate the preference structure of the opponent and uses a hill-climbing technique to explore the space of possible trade-offs for its own offers that are most likely to be accepted. The effectiveness of this method highly depends on the availability of prior domain knowledge that allows to determine the similarity of issue values. Coehoorn and Jennings [18] propose a method using Kernel Density Estimation for estimating the issue preferences of an opponent in multi-issue negotiations. It is assumed that the negotiation history is available and that the opponent employs a time-dependent strategy (refer to Subsection 2.1.3). The distance between successive counter-offers is used to calculate the opponent's issue weights and to assist an agent in making trade-offs in negotiation.

Some approaches use Bayesian learning in automated negotiation. For instance, Zeng et al. [90] use a Bayesian-learning representation and updating mechanism to model beliefs about the negotiation environment and the participating agents under a probabilistic framework; more precisely, they aim at enabling an agent to learn the reservation value of its opponent in single-issue negotiation. Another approach based on Bayesian learning is presented in [47]. Here the usage of a reasoning model based on a decision-making and belief-update mechanism is proposed to learn the likelihood of an opponent's profile; thereby it is assumed that the set of possible opponent profiles is known a priori. Hendrikx and Tykhonov [32] present a framework

⁴This review of heuristic approaches also appeared in our work [16].

for learning an opponent's preferences by making assumptions about the preference structure and rationality of its bidding process. It is assumed that (i) the opponent starts with optimal bids and then moves towards the bids close to the reservation value, (ii) its target utility can be expressed by a simple linear decreasing function, and (iii) the issue preferences (i.e., issue weights) are obtainable on the basis of the learned weight ranking. Moreover, the basic shape of the issue evaluation functions is restricted to downhill, uphill or triangular. In order to further reduce uncertainty in high-dimensional domains, issue independence is assumed to scale down the otherwise exponentially growing computational complexity. Oshrat et al. [60] developed a negotiating agent for effective multi-issue multi-attribute negotiations with both human counterparts and automated agents. The successful negotiation behavior of this agent is, to a large extent, grounded in its general opponent modeling component. This component applies Kernel Density Estimation to a collected database of past negotiation sessions for the purpose of estimating the probability of an offer to be accepted, the probability of the other party to propose a bid, and the expected averaged utility for the other party. The estimation of these values plays a central role in the agent's decision making. While the agent performs well, the approach taken is not suited for the type of negotiation we are considering (real-time, no prior knowledge, etc.) because opponent modeling is done off-line and requires knowledge about previous negotiation traces.

Strategy learning

Other available approaches aim at learning the negotiation strategy and decision model used by the opposing negotiator. For instance, Saha et al. [70] apply Chebyshev's polynomials to estimate the chance that an opponent accepts an offer relying on the decision history of its opponent. This work deals with single-issue negotiation, where an opponent's response can only be an accept or a reject. Brzostowski and Kowalczyk [10] investigate online prediction of future counter-offers on the basis of the past negotiation exchanges by using differentials. They assume that there mainly exist two independent factors that influence the behavior of an opposing agent, namely, time and imitation. The opponent is assumed to apply a weight combination of time- and behavior-dependent strategy (which we already discussed in Section 2.1.3). Hou [33] presents a learning mechanism that employs non-linear regression to predict the opponent's decision function in a single-issue negotiation setting. Thereby it is assumed that the opponent behavior can only be time-, behavior- or resources-dependent (with decision functions as proposed in [23]). In [11] an artificial neural network (ANN) is constructed with three layers that contain 52 neurons to model a negotiation process in a specific domain. The network exploits information about

past counter-offers to simulate future counter-offers of opponents. The training process requires a very large sample database and therefore cannot be applied in online mode.

Recently there is a growing body of work dealing with complex negotiations (as characterized in this thesis) by strategy learning. In [82] Gaussian processes are used to optimize an agent's own concession rate by predicting the maximal concession that the opponent is expected to make in the future. This approach, known as IAMhaggler2011, made the third place in ANAC 2011 [38]. Another approach based on Gaussian processes is described in [12], where a variant of the Gaussian processes regression model is used to alleviate the computational complexity of modeling opponents. Hao and Leung [28] propose another successful strategy, which was the winner of ANAC 2012. This method attempts to avoid concession as much as possible by adjusting the so-called non-exploitation time point. Moreover, it employs reinforcement-learning to improve the acceptance probability of its own proposals. These approaches (and several others) are used in subsequent chapters to evaluate our own approaches.

2.3.2 Game Theoretic Approaches

Game theory is the study of strategic behavior and interaction among autonomous agents. It has been applied in disciplines as diverse as economics, biology, linguistics and computer science [71]. Basically, there are two branches of game theory that are relevant to (automated) negotiation – cooperative and non-cooperative game theory. Cooperative game theory considers games in which participants are likely to form coalitions, or alternatively, games in which the basic unit is a group of agents that together achieve some joint payoff dependent on the contributions of group members. Non-cooperative game theory [71, 77] instead concentrates on strategies that can be used by self-interested players in order to maximize their individual profit.

In general, game theoretic techniques can be used to address two issues in agent-based negotiation scenarios [39]: (1) the design of protocols that govern the interactions between the negotiating parties; and (2) the design of an agent's strategy that it will apply in order to increase its own welfare. While proven to be useful and helpful in a variety of disciplines, there are two critical issues associated with the application of game theory to automated negotiations (see also [89]). One issue concerns the strong assumptions made about the negotiation settings. For instance, game theoretic approaches usually assume that it is possible to characterize an agent's preferences with respect to all possible outcomes. However, finding or defining those preferences is challenging and may even be impossible for some settings. Furthermore, these approaches often assume unbounded computational capability, which means that no computational/time cost needs to be considered in search of mutually accepted solu-

tions within the range of possible outcomes, even if the outcome space is extremely large.

Despite these critical issues, game theory offers useful concepts to characterize the solution space of negotiation. For example, a well known and important concept is the Nash Equilibrium, where neither participant can benefit by choosing an alternative action, given that all other participants do not change their action [58]. Another famous example is the concept of Pareto optimality, which can be used as a good indicator for the efficiency of a specific solution; thereby, an outcome is said to be Pareto optimal if no negotiation participant can be better off without making at least one other participant worse off. There are also some useful concepts developed in game theory for characterizing the fairness of a negotiated solution, such as the utilitarian [56] and the Kalai-Smorodinsky [42] solution.

2.3.3 Argumentation-based Approaches

Argumentation-based negotiation [39, 65, 73], which is another research line in automated negotiation, aims at providing additional information exchange between negotiators other than the “pure”, complete trading proposals (e.g., an offer or counter-offer). This additional information comes in form of arguments that support or criticize a proposal made by an agent. A major advantage of exchanging extra information (in addition to offers) is to enable agents to reflect on received proposals as well as to explain their negotiation partners why they are not satisfied with a received proposal instead just rejecting a proposal without any reasons. The ultimate goal of argumentation-based negotiation is to provide convincing arguments that lead to a change in the preferences of the respective other negotiator and thus arrive at an agreement between participants.

A difficulty with this form of negotiation is the possibly significant computational overhead needed for sophisticated communication (this concerns both the protocol and language to be used) as well as the complex knowledge-based reasoning processes needed for constructing and evaluating arguments. Moreover, further difficulty arises due to the fact that agents may intentionally provide arguments that are not truthful in order to mislead their negotiation partners. In response to this additional methods, at the cost of more computation and communication efforts, must be taken to evaluate an argument’s credibility [39]. Argumentation-based negotiation till now is still widely unexplored and is still facing a host of open questions, it is thus not within the focus of this thesis.

2.3.4 Automated Negotiating Agent Competition

The Automated Negotiating Agent Competition (ANAC) [37, 38, 52] is a yearly international competition (in conjunction with the International Conference on Autonomous Agents and Multi-agent Systems). This event was jointly initialized by Delft University of Technology and Bar-Ilan University in 2010 for fostering the development of practical negotiating agents that are able to proficiently negotiate against unknown opponents. With a large number of state-of-the-art negotiating agents and negotiation domains, ANAC provides a useful benchmark for objectively evaluating negotiation strategies. ANAC employs the Generic Environment for Negotiation with Intelligent multi-purpose Usage Simulation (Genius) [30] as the official test platform. This framework supports negotiation sessions where the behavior and preferences of opponents are unknown and where the negotiation sessions are subject to discounting effects and real-time constraints. Genius also allows to compare new negotiation strategies against various state-of-the-art negotiation agents that have been implemented within this framework.

In a competition, each agent plays against other agents in every considered domain, where the two agents involved in a negotiation act in turn in conflicting roles (e.g., “buyer” and “seller”). Let Λ be a set of negotiating agents, and let a and b be two agents in this set. Suppose that a and b negotiate in some domain $D \in \mathcal{D}$, where \mathcal{D} is a given set of predefined domains. Let the two roles to be played by a and b in domain D be denoted by P_1^D and P_2^D , and let $U_{a \rightarrow b}^D(a, P_1^D)$ denote the score of agent a in a session that was opened by a playing role P_1^D . Then, the score of agent a , when playing the role P_1^D , is calculated as

$$S(a, P_1^D) = \frac{1}{2(|\Lambda| - 1)} \sum_{b \in \{\Lambda \setminus a\}} (U_{a \rightarrow b}^D(a, P_1^D) + U_{b \rightarrow a}^D(a, P_1^D)) \quad (2.3)$$

and the score of a , when playing the other role P_2^D , is given by

$$S(a, P_2^D) = \frac{1}{2(|\Lambda| - 1)} \sum_{b \in \{\Lambda \setminus a\}} (U_{a \rightarrow b}^D(a, P_2^D) + U_{b \rightarrow a}^D(a, P_2^D)) \quad (2.4)$$

The overall score of agent a for domain D is then given by

$$S(a, D) = \frac{S(a, P_1^D) + S(a, P_2^D)}{2} \quad (2.5)$$

and the final score achieved by a over all considered domains is

$$S(a) = \frac{1}{|\mathcal{D}|} \sum_{D \in \mathcal{D}} S(a, D) \quad . \quad (2.6)$$

Part II

Design of Automated Negotiation Strategies

3

Strategies based on Decomposition Techniques

This chapter is based on: *S.Chen and G.Weiss. OMAC: A Discrete Wavelet Transformation based Negotiation Agent. Novel Insights in Agent-based Complex Automated Negotiation (Studies in Computational Intelligence, Vol. 535, pp. 187-196). 2014. Springer.*

S. Chen and G. Weiss. An efficient and adaptive approach to negotiation in complex environments. In Proceedings of the 20th European Conference on Artificial Intelligence (ECAI, pp. 228-233). 2012. IOS Press.

S. Chen and G. Weiss. A novel strategy for efficient negotiation in complex environments. In Proceedings of the 10th German Conference on Multiagent System Technologies (MATES, pp. 68-82). Lecture Notes in Computer Science, Vol. 7598. 2012. Springer-Verlag.

This chapter presents two automated negotiation strategies based on decomposition techniques. First, OMAC¹, which learns opponent behavior by means of discrete wavelet transformation, is given in Section 3.1. Next, Section 3.2 proposes an alternative strategy (EMAR²) depending on a more adaptive learning scheme. We also evaluate the empirical performance evaluation (i.e. tournament-based) of the two strategies, respectively.

3.1 OMAC

OMAC includes two core stages – opponent modeling and decision-making mechanism

¹OMAC stands for Opponent Modeling and Adaptive Concession, representing two key aspects of successful negotiation: efficient opponent modeling and adaptive concession making.

²EMAR stands for Empirical Mode Decomposition (EMD, e.g. [36]) and Autoregressive Moving Average (ARMA, e.g. [9]).

– as described in detail in Sections 3.1.2 and 3.1.3, respectively. An overview of OMAC is firstly given in *Algorithm 1*. Before diving into the details of OMAC, we first introduce some related respects of discrete wavelet transformation, which plays a central role in learning opponent for OMAC.

Algorithm 1 The OMAC strategy. t_c refers to the current time, δ the time discounting factor, λ the layer of wavelet decomposition, ψ the wavelet function, and t_{max} the deadline of negotiation. O_{opp} is the latest offer of the opponent, and O_{own} the offer to be proposed by OMAC. χ represents the time series comprised of the maximum utilities over intervals. Let v be the smooth component of λ -th order wavelet decomposition based on ψ , and α the predicted main tendency of χ . t_l is the time we perform prediction process and u_l is the utility of our most recent offer. u' is the target utility at time t_c . R is the reserved utility function.

```

1: Require :  $t_{max}, \delta, \lambda, \psi, R$ 
2: while  $t_c \leq t_{max}$  do
3:    $O_{opp} \leftarrow \text{receiveMessage}()$ ;
4:    $\text{recordBids}(t_c, O_{opp})$ ;
5:   if  $\text{needUpdate}(t_c)$  then
6:      $\chi \leftarrow \text{preprocessData}(t_c)$ 
7:      $(\alpha, t_l, u_l) \leftarrow \text{predict}(\chi, \lambda, \psi)$ ;
8:   end if
9:    $u' = \text{getTarUtility}(t_c, t_l, u_l, \delta, \alpha, R)$ ;
10:  if  $\text{getOwnUtility}(O_{opp}, t_c, \delta) \geq u'$  then
11:     $\text{accept}(O_{opp})$ ;
12:  else
13:     $O_{own} \leftarrow \text{constructOffer}(u')$ ;
14:     $\text{proposeBid}(O_{own})$ ;
15:  end if
16: end while

```

3.1.1 Discrete Wavelet Transformation

Discrete wavelet transformation (DWT) is a type of multi-resolution wavelet analysis that provides a time-frequency representation of a signal and, based on this, it is capable of capturing time localizations of frequency components. DWT has become increasingly important and popular as an efficient multi-scaling tool for exploring features. This is due to the fact that it can offer with modest computational effort high-quality solutions (with complexity $O(n)$) to non-trivial problems such as feature extraction, noise reduction, function approximation and signal compression. OMAC employs DWT to extract the main trend of the opponent's concession over time from its previous counter-offers. In the following, aspects of DWT are described that are relevant to OMAC; further details can be found in, e.g., [20, 69].

In DWT a time-frequency representation of a signal is obtained through digital filtering techniques, where two sets of functions are utilized: scaling functions using low-pass filters and wavelet functions using high-pass filters. More precisely, a signal is passed through a series of high pass filters to analyze the high frequencies, and similarly it is passed through a series of low pass filters to analyze the low frequencies. In so doing, DWT decomposes a signal into two parts, an approximation part and a detail part. The former is smooth and reveals the basic trend of the original signal, and the latter is rough and in general corresponds to short-term noise from the higher-frequency band.

The decomposition process can be applied recursively as follows:

$$\begin{aligned} y_{low}[k] &= \sum_n f[n] \cdot h[2k - n] \\ y_{high}[k] &= \sum_n f[n] \cdot g[2k - n] \end{aligned} \quad (3.1)$$

with $f[n]$ being the signal, $h[n]$ a halfband high-pass filter, $g[n]$ a halfband low-pass filter, and $y_{low}[k]$ and $y_{high}[k]$ the outputs of the low-pass and high-pass filters, respectively. The iterative application of DWT results in different levels of detail of the input signal; in other words, it decomposes the approximation part into a “further smoothed” component and a corresponding detail component. The further smoothed component contains longer period information and provides a more accurate trend of the signal. For instance, f can firstly be decomposed into a rough smooth part (a_1) and a detail part (d_1), and then the resulting part a_1 can be decomposed in finer components, that is, $a_1 = a_2 + d_2$, and so on. This iterative process is captured by the below diagram:

$$\begin{array}{ccccccc} f & \cdots & a_1 & \cdots & a_2 & \cdots & a_3 & \cdots & a_n \\ & & \ddots & & \ddots & & \ddots & & \\ & & d_1 & & d_2 & & d_3 & \cdots & d_n \end{array}$$

where a_1, a_2, \dots, a_n are the approximation parts and d_1, d_2, \dots, d_n are the detail parts of f . Based upon this recursive process, the signal can be expressed as $f = d_1 + d_2 + \dots + d_n + a_n$

Daubechies wavelets are a family of orthogonal wavelets defining a DWT. They are applied for solving a range of problems, e.g., self-similarity properties of a signal, fractal problems, signal discontinuities, etc. The results reported in this chapter are achieved through wavelet decomposition using the Daubechies’ wavelets of order 10 (referred to as “DB10” afterwards). We use the notation below to represent the

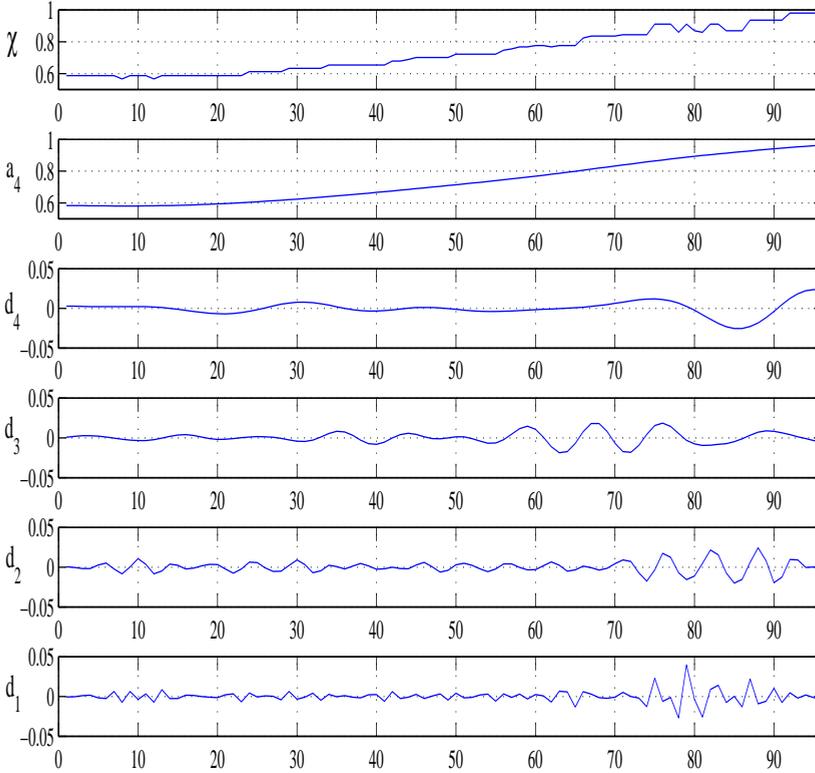


Figure 3.1: 4-level decomposition of the utilities series χ obtained from IAMhag-gler2011 in the negotiation domain *Airport site selection* using DB10. The vertical axis shows score and the horizontal axis represents the percentage of time (%). The 4-level DWT decomposes the original input χ into five sub-components including $d_1 \dots d_4$ (detail parts) and a_4 (approximation part). The resulting components are shown below χ , their frequency increases from d_1 to d_4 (i.e., they are more and more smooth). The final approximation part – a_4 is the long-term trend of χ .

decomposition relation in our case:

$$\chi = v + \sum_{n=1}^{\lambda} d_n \quad (3.2)$$

where χ is the time series (i.e., the maximum utilities of counter-offers over intervals,

refer to Section 3.1.2), v represents the n -layer approximation component of χ (i.e., a_n), d_n is the n -layer detail part, and λ the number of decomposition level.

A concrete example of applying DWT in negotiation is given in Figure 3.1, which shows the curve of the received utilities (i.e., the original signal) in the domain *Airport site selection* when negotiating with the agent IAMhaggler2011 (more information about domains and agents can found in later sections). The decomposition results were obtained with $\lambda = 4$. The curve at the top of figure represents χ , d_n is the detail component of the n^{th} decomposition layer and a_4 is the approximation on the final layer (i.e., the fourth), corresponding to ω in Equation 3.2. This figure clearly shows that a_4 is a pretty good approximation of the main trend of the original signal. As can be seen, the noise/variation represented by those detail components (e.g., d_1 to d_4) is irrelevant to its trend.

3.1.2 Opponent Modeling

According to OMAC, the aim of opponent modeling realized by a negotiating agent is to estimate the utilities of future counter-offers it will receive from its opponent. This corresponds to the lines 3 to 8 in *Algorithm 1*. Opponent modeling is done through a combination of wavelets analysis and cubic smoothing spline. When receiving a new bid from the opponent at the time t_c , the agent records the time stamp t_c and the utility $U(O_{opp})$ this bid has according to the agent's utility function. The maximum utilities in consecutive equal time intervals and the corresponding time stamps are used periodically as basis for predicting the opponent's behavior (lines 5 and 6). The reasons for a periodical updating are similar to those mentioned in [82]. Firstly, this degrades the computation complexity so that the agent's response time is kept low. Assume that all observed counter-offers were taken as inputs, then the agent might have to deal with thousands of data points in every single session. This computational load would have a clear negative impact on the quality of negotiation in a real-time constraint setting. Secondly, the effect of noise can be reduced. In multi-issue negotiation a small change in utility of the opponent can result in a large utility change for the negotiator and this can easily result in a misinterpretation of opponent's behavior.

Behavior prediction is mainly done by applying DWT to the time series χ ; this is captured by line 7. OMAC focuses on the approximation part and intentionally ignores the detail part for three reasons. First, the approximation part represents the trend of the opponent concession in terms of utility and indicates how the concession of opponent will develop in the future. Second, it is smooth enough (compared to the original signals, i.e. χ) to allow for quality prediction performance. Third, the detail part contains information which is of little value in a negotiation setting. As we saw

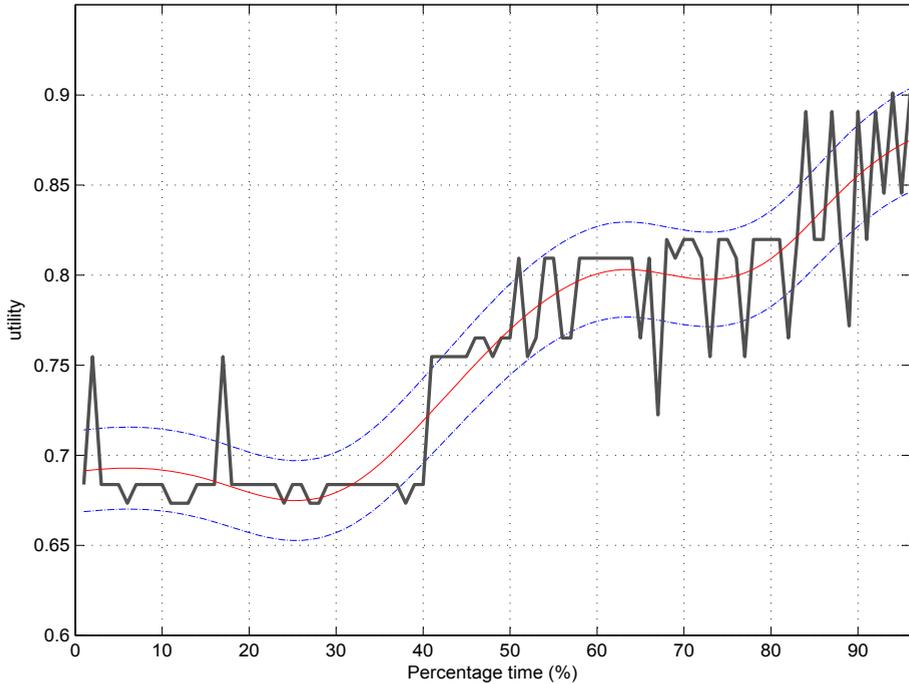


Figure 3.2: Illustrating the opponent’s concession (given by χ , the thick solid line) and the corresponding approximation part v (the thin solid line) when negotiating with Agent_K2 in the *Camera* domain. The two dash-dot lines represent the estimated upper and lower bounds of χ . (Details of the agents and domains are given in Section 3.1.4.)

in various empirical investigations, the ratio between the main tendency term and the original signal tends to be about 0.98 with a small standard deviation. Precise extension of those detailed components can improve effectiveness of our model slightly, it is however very costly for a medium-range lead time in real-time negotiation. An example can be found in Figure 3.2 which shows χ and its corresponding approximation part v along with the estimated upper and lower bounds of χ . The two bounds are represented by $v \pm \sigma$, where σ is the standard deviation of the ratio between χ and v .

In order to forecast the opponent’s future behavior, cubic smoothing spline is used to extend the smooth component v . Cubic spline is widely used as a tool for prediction, see [86]. For equally spaced time series, a cubic spline is a smoothing

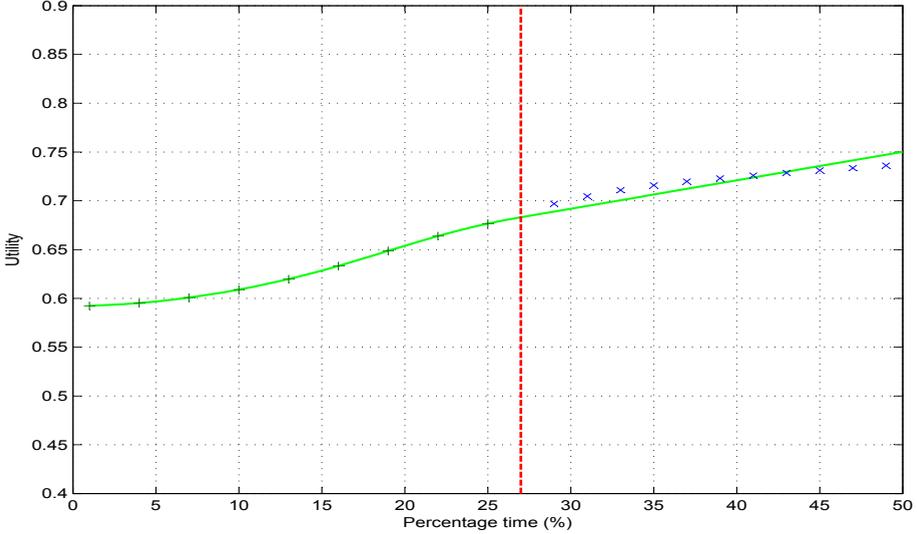


Figure 3.3: Illustration of the predictive power of our approach in two consecutive ranges. The dash line indicates the time point t_c at which the current prediction is made. The plus signs on left of the dash line are the actual points of v before t_c . The crosses to the right of the dash line show the actual points of v after t_c . The extended version of $v - \alpha$ (i.e., the prediction of v) is shown by the solid line. These results are achieved for agent Iamhaggler2011 in the domain *Amsterdam party*.

piecewise function, denoted as the function $\hat{g}(t)$ which minimizes:

$$p \sum_{t=1}^n w(t)(f(t) - \hat{g}(t))^2 + (1 - p) \int (\hat{g}(u)'')^2 du \quad (3.3)$$

where p is the smoothing parameter controlling the rate of exchange between the residual error described by the sum of squared residuals and local variation represented by the square integral of the second derivative of g and w is the weight vector (for further details, refer to [21]).

Figure 3.3 shows the actual and the predicted smooth parts of opponent concession at different time points for the opponent Iamhaggler2011: as this figure illustrates, cubic spline is able to forecast the given signal within a medium range very well. Since OMAC applies a periodical updating mechanism, it is not necessary and not wise to forecast globally (i.e., from the current moment to the end point of negotiation), because this probably brings too much noise into the prediction. OMAC limits the range of forecasting to ζ intervals and in this way achieves efficiency and noise reduction.

3.1.3 Decision-making Mechanism

Given the extended version of the smooth part $-\alpha$, we now discuss how to use it for adaptively setting the concession rate of our expected utility (see line 9 in *Algorithm 3*). A possibility is to maximize the expected utility merely according to the predicted opponent move. This is quite straightforward but may be not so effective. Suppose the negotiation partners are “tough” and always avoid making any concession in bargaining. In this case the result of prediction could indicate a very low expectation about the utility offered by the opponent and this, in turn, would result in an adverse concession. In our approach a simple function R , called reserved utility function, is used to realize concession adaptation. This function guarantees the minimum utility at each given time step. This is because the function values are set as the lower bound of our expected utilities. Moreover, in principle it makes concession over time, thereby taking into account the impact of the discounting factor. Specifically, the reserved utility function is given by:

$$R(t) = u_{res} + (1 - t^{1/\beta})(\maxUtility(p) \cdot \delta^\eta - u_{res}) \quad (3.4)$$

where u_{res} is the minimum utility the agent would accept, β is a parameter which has a direct impact on the concession rate, $\maxUtility(p)$ is the function specifying the maximum utility given by the preference profile p of a negotiation domain, and η is a parameter called risk factor which reflects the agent’s expectation about the maximum utility it can achieve.

We define the estimated received utility $E_{ru}(t)$, which gives our agent the expectation of opponent’s future concession, as follows:

$$E_{ru}(t) = D(\alpha(t)(1 + Stdev(ratio_{[t_b, t_c]})), t), \quad t \in [t_c, t_s] \quad (3.5)$$

where $Stdev(ratio_{[t_b, t_c]})$ is the standard deviation of ratio between the smooth part v and the original signal χ from the beginning of negotiation(t_b) till now and t_s is the end of α .

Suppose the future expectation the agent has obtained from $E_{ru}(t)$ is optimistic, in other words, there exists an interval $\{T|T \neq \emptyset, T \subseteq [t_c, t_s]\}$, so that

$$E_{ru}(t) \geq D(R(t), t), \quad t \in T \quad (3.6)$$

OMAC then sets the time \hat{t} at which the optimal estimated utility \hat{u} is reached as:

$$\hat{t} = \operatorname{argmax}_{t \in T} (E_{ru}(t) - D(R(t), t)) \quad (3.7)$$

and \hat{u} is simply assigned by:

$$\hat{u} = E_{ru}(\hat{t}) \quad (3.8)$$

When the opponent's future concession is estimated to be below the agent's expectations according to $R(t)$ (i.e., there is no such interval T described above), OMAC investigates whether the best possible outcome under that "pessimistic" expectation of opponent concession should be accepted given the threshold ρ . This outcome is denoted as ξ and is given by:

$$\xi = \rho^{-1} \cdot E_{ru}(t_\xi) / D(R(t_\xi), t_\xi), \quad t_\xi \in [t_c, t_s] \quad (3.9)$$

where ρ is the tolerance threshold to accept $E_{ru}(t_\xi)$ as target utility and t_ξ is given by:

$$t_\xi = \operatorname{argmin}_{t \in [t_c, t_s]} (|E_{ru}(t) - D(R(t), t)|) \quad (3.10)$$

The rationality behind it is that if the agent rejects the "locally optimal" counter-offer, the agent will probably lose the opportunity to reach a "globally good" agreement (especially in discounting domains). If $\xi > 1$, \hat{u} and \hat{t} are assigned to $E_{ru}(t_\xi)$ and t_ξ , respectively. Moreover, the agent records the utility and time of its last bid as u_l and t_l , respectively. Otherwise, the estimated utility is set to -1, meaning it does not take effect anymore, and $D(R(t_c), t_c)$ is used to set the target utility u' .

When the agent expects to achieve better outcomes (see Equation 3.6), the optimal estimated utility \hat{u} is chosen as the target utility for our agent's future bids. Obviously, it is not rational to concede immediately to \hat{u} when $u_l \geq \hat{u}$, nor should it shift to \hat{u} without delay given $u_l < \hat{u}$, especially because the predication may be not absolutely accurate. To simplify the strategy, OMAC applies a linear concession making and the concession rate is dynamically adjusted to grasp every chance to maximize its profit. Overall, the target utility u' is given as follows:

$$u' = \begin{cases} D(R(t), t) & \text{if } \hat{u} = -1 \\ \hat{u} + (u_l - \hat{u}) \frac{t - \hat{t}}{t_l - \hat{t}} & \text{otherwise} \end{cases} \quad (3.11)$$

The response stage corresponds to lines 10 to 15 in *Algorithm 1*. With the target utility u' known (Equation 3.11), the agent then needs to examine the counter-offer to see if the utility of that offer $U(O_{opp})$ is higher than the target utility. If so, it accepts this counter-offer and, with that, terminates the negotiation session. Otherwise, the agent construct a bid to be proposed next round whose utility is indicated by u' .

In multi-issue negotiation, offers with exactly the same utility for one side can have different values for the other party. Moreover, in time-limited negotiation scenarios no explicit limitation is imposed on the number of negotiation rounds and it is possible

to generate many offers having a utility close to u' . OMAC takes advantage of this and aims at generating many offers in order to explore the space of possible outcomes and to increase the acceptance chance of own bids. Specifically, offers are constructed in such a way that the agent randomly selects an offer whose utility is in the range $[0.99u', 1.01u']$. If no such solution is found, the latest offer made by the agent is used again in the subsequent round. Moreover, in view of negotiation efficiency, if u' drops below the utility of the best counter-offer according to the agent's utility function, this best counter-offer is proposed by the agent as its next offer. This makes sense because the counter-offer tends to satisfy the expectation of opponent and is thus likely to be accepted by the opponent.

3.1.4 Experimental Analysis

The performance evaluation of OMAC is done with GENIUS [30] which is also used as a competition platform for ANAC. It allows to compare agents (representing different negotiation strategies) across a variety of application domains under real-time constraints, where the preference profiles of two negotiating agents are specified for the individual domains.

Experimental settings of experiments

OMAC-agent is compared against the top five agents of ANAC 2011 (OMAC-agent refers to the agent which applies OMAC).³ Moreover, we use five application domains created for ANAC, all of which are originally non-discounting except *Camera*. We refer to the non-discounting domain *Travel* as U_1 , *Amsterdam party* as U_2 , *England vs Zimbabwe* as U_3 , *IteX vs Cypress* as U_4 , *Camera* as U_5 . The corresponding domains with time-dependent discounting factor are referred to as D_1, D_2, \dots, D_5 , respectively. Three of these domains were used in ANAC 2010, and the others have been selected by ANAC 2011. This choice of domains from ANAC2010 and 2011 makes the overall setting balanced (a wide range of domain characteristics are covered) and fair, and avoids any advantageous bias for OMAC resp. OMAC-agent. (With respect to fairness, note that the developers of the 2011 winners knew the ANAC2010 domains and thus had the opportunity to optimize their agents accordingly). For each of the ten domains, we run a tournament consisting of 6 agents 10 times to get results with high statistical confidence, where each agent repeats the negotiation against all other negotiators playing different roles in turn. In competition, agents will not be

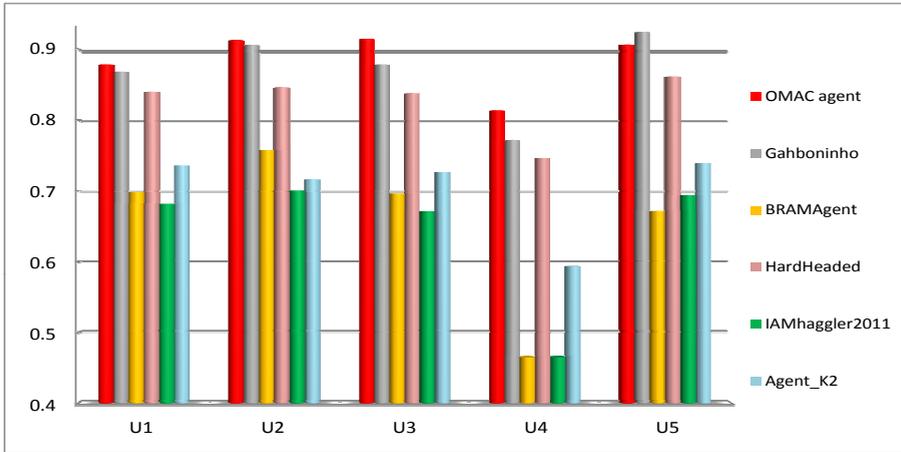
³The sets of opponents used in the tournaments for each proposed strategy include those strategies that are most close to the negotiation design and evaluation objectives considered in this chapter. In Chapters 5 & 6 we even use a somewhat different set of opponents, composed of the top-performing agents from ANAC 2012 & 2013, to compare all proposed strategies in a wide range of scenarios.

given any information about the opponents’ strategies or other private information. Furthermore, they are prohibited from taking advantage of previous encounters with their opponents. The time limit for every single negotiation session is 180 seconds in total (to be shared by the two negotiators). In the implementation of OMAC-agent, ρ is set to 0.95 and the overall duration of a session is dividend into 100 consecutive intervals of 1.8 seconds each. In addition, the maximum predictive range ζ is limited to 15 intervals, the risk factor η is set to 0.2 and ψ is Daubechies’ wavelets of order 10 with λ being 4. (In our experiments we found out that OMAC works well for a very broad range of parameters.)

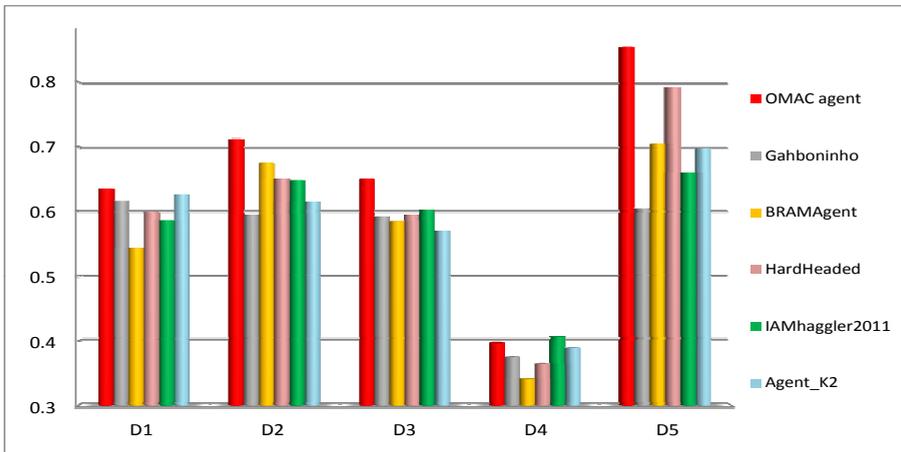
Experimental outcomes

Figure 3.4 depicts experimental results based on non-discounting domains (3.4(a)) and discounting domains (3.4(b)). As these figures show, OMAC-agent demonstrates excellent bargaining skills in each of the ten domains. More specifically, the agent finished first in eight domains and finished second in the remaining two domains (U_5 and D_4) with a very small disadvantage (about 2.35% at most). The domains used for testing cover a variety of domains in terms of three significant aspects. First, the outcome space ranges from large (U_1 with 188,160) over medium (U_2 having 3,024) to small (U_4 with 180). Second, the opposition of domains (i.e. how compatible the interests of two parties are w.r.t. obtaining a joint benefit) has a broad spectrum, ranging from weak (U_1) over medium (U_3) to strong (U_4). Third, the domains show a broad spectrum of discounting factors, ranging from 0.4 (for D_1) to 1.0 (for all non-discounting domains). The distinct discounting factors (from the highest D_1 to non-discounting domains, i.e. U_n) of the ten domains require agent’s appropriate decision-making under real-time pressure against unknown opponents. Together these aspects make the used domains very challenging. In view of these aspects the obtained results clearly indicate that OMAC is not limited to usage in a few specific domains, but is of value for a broad range of applications.

Table 3.1 shows the average normalized scores for each agent in the discounting and non-discounting domains, and Table 3.2 shows for each agent the overall raw and normalized scores averaged over the ten domains. As usual, the scores were standardized by using the maximum and minimum utility obtained by all participants in the domain. According to Table 3.1, where the agents are ordered by the final ranking shown in Table 3.2, OMAC-agent completely outclassed others with a small variance. Another interesting observation is that Gahboninho performed very well in non-discounting domains (closely following OMAC-agent), while it achieved the worst score in discounting domains. A similar performance behavior was shown by lamhaggler2011. This means that their behaviors are somewhat “extreme” and not



(a) Average raw scores of all agents in all domains without discounting effect



(b) Average raw scores of all agents in all domains with discounting effect

Figure 3.4: Average raw scores of all agents in ten domains. The vertical axis shows utility and horizontal axis shows domain.

adaptive enough to cope with the given range of negotiation domains. Compared to these agents, the champion of ANAC 2011 – Hardhead – achieved a more balanced and effective behavior.

From Table 3.2 it can be seen that the best overall performance is achieved by OMAC-agent – its utility (in terms of the normalized score) is 28.1% higher than the average utility of the other agents. The second best performer is Hardheaded, with a disadvantage of about 13% compared to OMAC-agent, and the lowest performance was shown by Iamhaggler2011.

Agent	Non-discounting domain		Discounting domain	
	mean	variance	mean	variance
OMAC-agent	0.831	0.0008	0.714	0.0016
HardHeaded	0.741	0.0010	0.623	0.0013
Gahboninho	0.807	0.0023	0.555	0.0031
Agent_K2	0.557	0.0025	0.591	0.0023
BRAMAgent	0.502	0.0089	0.577	0.0084
Iamhaggler2011	0.474	0.0046	0.602	0.0021

Table 3.1: The average normalized score of agents for discounting and non-discounting domains.

Agent	Raw Score		Normalized Score	
	mean	variance	mean	variance
OMAC-agent	0.767	0.0013	0.772	0.0101
HardHeaded	0.713	0.0011	0.682	0.0035
Gahboninho	0.713	0.0027	0.681	0.0083
Agent_K2	0.641	0.0034	0.574	0.0059
BRAMAgent	0.613	0.0087	0.540	0.0208
IAMhaggler2011	0.612	0.0024	0.538	0.0088

Table 3.2: Overall performance of OMAC and benchmarking agents.

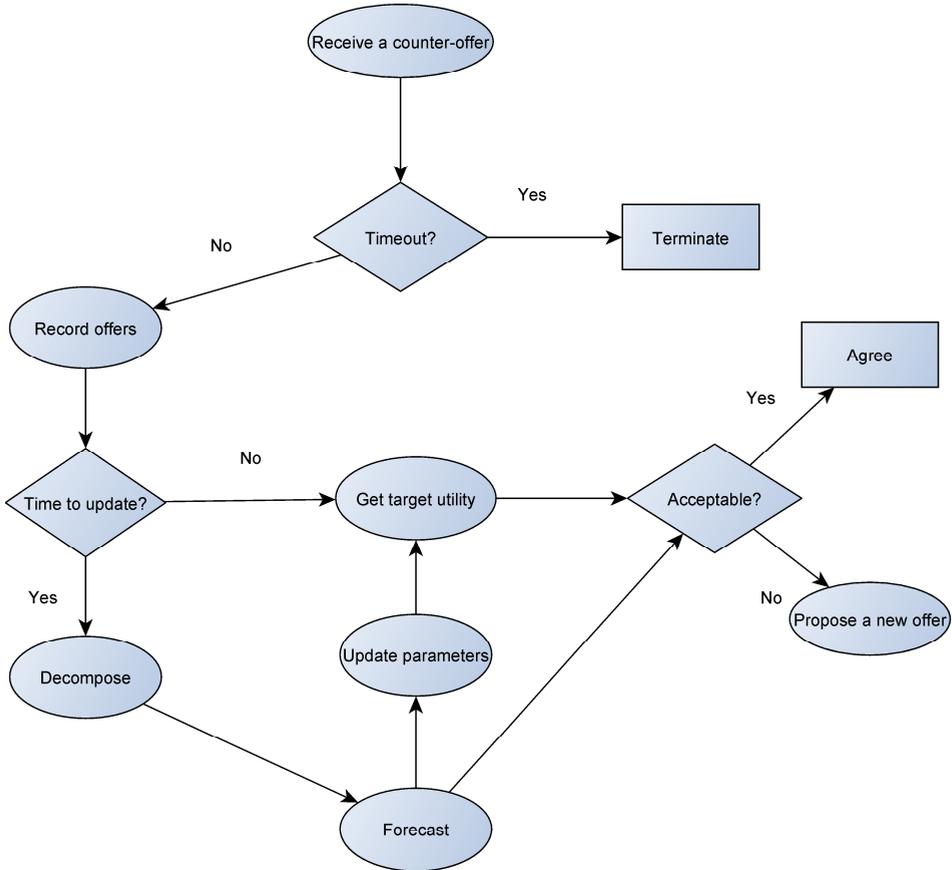


Figure 3.5: A brief flowchart of EMAR.

3.2 EMAR

EMAR also includes two core stages – opponent modeling and adaptive concession making – as described in detail in 3.2.1 and 3.2.2, respectively. An overview of EMAR is given in *Algorithm 2*, which for clarity is also visualized in Figure 3.5. The individual steps of *Algorithm 2* are explained in the following text.

3.2.1 Opponent Modeling

Opponent modeling realized by EMAR aims at predicting the future moves of the negotiating opponents. The process of opponent modeling corresponds to the lines 2 to 11 in *Algorithm 2*. When receiving a new bid from the negotiation opponent

Algorithm 2 The EMAR approach. Let t_c be the current time, δ the time discounting factor, t_{\max} the deadline of negotiation and ϑ the reservation value. O_{opp} is the latest offer of the opponent and O_{own} is a new offer to be proposed by EMAR. χ is the time series comprised of the maximum utilities over intervals. ξ is the lead time for prediction and ω is the estimated central tendency of χ . c_i is the i -th IMF component and r_n is the final residue (more details will be given in the following text). E is predicted received utility series. u_{res} is the dynamic reservation utility, specifying the lowest expectation to negotiation payoff, and e_{min} is the conservative estimation of opponent concession. R is the dynamic conservative expectation function. u' is the target utility at time t_c .

```

1: Require:  $R, \delta, \xi, \vartheta, t_{\max}$ 
2: while  $t_c \leq t_{\max}$  do
3:    $O_{\text{opp}} \leftarrow \text{receiveMessage}$ ;
4:    $\text{recordBids}(t_c, O_{\text{opp}})$ ;
5:   if  $\text{TimeToUpdate}(t_c)$  then
6:      $\chi \leftarrow \text{preprocessData}(t_c)$ ;
7:      $(c_0, \dots, c_n, r_n) \leftarrow \text{decompose}(\chi)$ ;
8:      $(\omega, E) \leftarrow \text{forecast}(c_0, \dots, c_n, r_n, \xi)$ ;
9:      $(u_{\text{res}}, e_{\text{min}}) \leftarrow \text{updateRes}(\omega, \chi, \vartheta, t_c)$ ;
10:     $R \leftarrow (u_{\text{res}}, e_{\text{min}})$ ;
11:   end if
12:    $u' = \text{getTarget}(t_c, E, \delta, R)$ ;
13:   if  $\text{isAcceptable}(u', O_{\text{opp}}, t_c, \delta)$  then
14:      $\text{accept}(O_{\text{opp}})$ ;
15:   else
16:      $O_{\text{own}} \leftarrow \text{constructOffer}(u')$ ;
17:      $\text{proposeNewBid}(O_{\text{own}})$ ;
18:   end if
19: end while

```

at the time t_c , the agent records the time stamp t_c and the utility $U(O_{\text{opp}})$ this bid offers according to its utility function. The maximum utilities in consecutive equal time intervals and the corresponding time stamps are used periodically as input for predicting the opponent's behavior (line 5 and 6). The reasons for *periodical* updating are twofold. First, this reduces the computation complexity of EMAR so that the response speed is improved. Assume *all* observed counter-offers were taken as input, then it would be necessary deal with perhaps many thousands of data points at once. This computational load would have a clear negative impact on the quality of negotiation in a real-time setting. Second, the effect of noise can be reduced. This is important because in multi-issue negotiations a small change in utility of the opponent can result in a large utility change for the other agent – and this can easily result in a fatal misinterpretation of the opponent's behavior.

The general idea behind opponent modeling realized by EMAR is to apply the “divide-and-conquer” principle to construct a reasonable forecasting methodology. Opponent modeling is mainly based on a combination of Empirical Mode Decomposition (EMD) [25, 34, 36] and Autoregressive Moving Average (ARMA) [9]. In more detail, EMD is first employed to decompose the time series given by the utilities of past counter-offers into a finite number of components in order to make the prediction task simplified, and then ARMA is applied to predict future values of these sub-components.

EMD, which is introduced by the Hilbert-Huang transform (HHT), is a decomposition technique which relies on time-local characteristics of data and can deal with nonlinear and non-stationary time series in a adaptive manner. It has been widely applied as a powerful data analysis tool in a broad scope of fields such as finance, image processing, ocean engineering and solar studies. A main advantage of EMD as a decomposition method is that it is very suitable for analyzing complicated data and is fully data driven (thus requiring no additional decomposition information) – this makes EMD an effective and efficient decomposition method. Compared to traditional Fourier and wavelet decompositions, EMD has several distinct advantages [35, 87]. First of all, fluctuations within a time series are automatically selected from the time series. Second, EMD can adaptively decompose a time series into several independent components called Intrinsic Mode Functions (IMFs). With the usage of the IMFs a residue can be calculated which captures the main trend of the time series. Lastly, unlike wavelet decomposition, no filter base function need to be determined beforehand – which is particularly helpful when there is no prior knowledge about which filters work properly.

Any data series can be decomposed into IMFs according to the following sifting procedure (let $k \geq 1$, k indicates the iterative decomposition level):

1. Take signal r_{k-1} as input, with r_0 representing the original signal $\chi(t)$.
 - (a) Identify all local extrema of the signal r_{k-1} .
 - (b) Construct the upper envelope $\text{Upp}(r_{k-1})$ and the lower envelop $\text{Low}(r_{k-1})$ by interpolating via cubic spline the maximum and minimum values, respectively.
 - (c) Approximate the local average by the envelope mean $\text{Mean}(r_{k-1}) = (\text{Upp}(r_{k-1}) + \text{Low}(r_{k-1})) / 2$.
 - (d) Compute the candidate implicit mode $h_{kn} = r_{k-1} - \text{Mean}(r_{k-1})$.
 - (e) If h_{kn} is an IMF, then calculate r_k as $r_k = r_{k-1} - h_{kn}$. Otherwise replace r_{k-1} with h_{kn} and repeat sifting.
2. If r_k has an implicit oscillation mode, set r_k as input signal and repeat step 1.

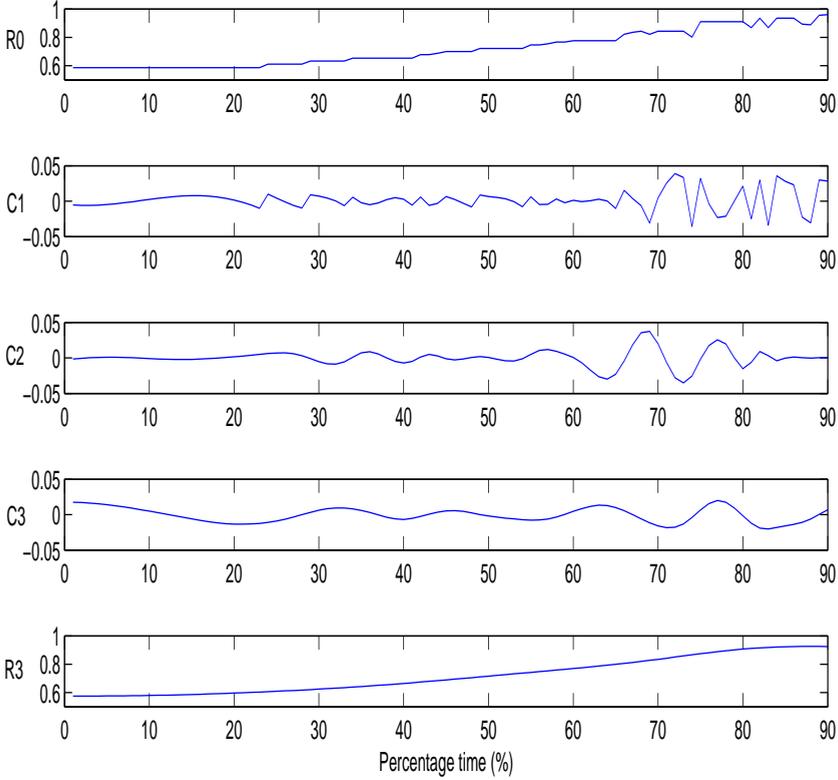


Figure 3.6: Illustrating the resulting components achieved by the application of EMD, where R_0 is original signal, the i -th IMF is given as C_i and the residue is shown by R_3 . R_0 is offered by the agent Iamhaggler2011 in domain *England vs Zimbabwe*. Details of agents and domain are given in Section 3.2.3.

This sifting process serves two purposes: to eliminate riding waves and to make the wave profiles symmetric.

The decomposition procedure can be repeated on all subsequent components r_j , and the result is

$$r_0 - c_1 = r_1, r_1 - c_2 = r_2, \dots, r_{n-1} - c_n = r_n. \quad (3.12)$$

This procedure terminates when (1) the latest residue r_k becomes a monotonic function (from which no more IMFs can be extracted) or (2) the IMF component c_k or the

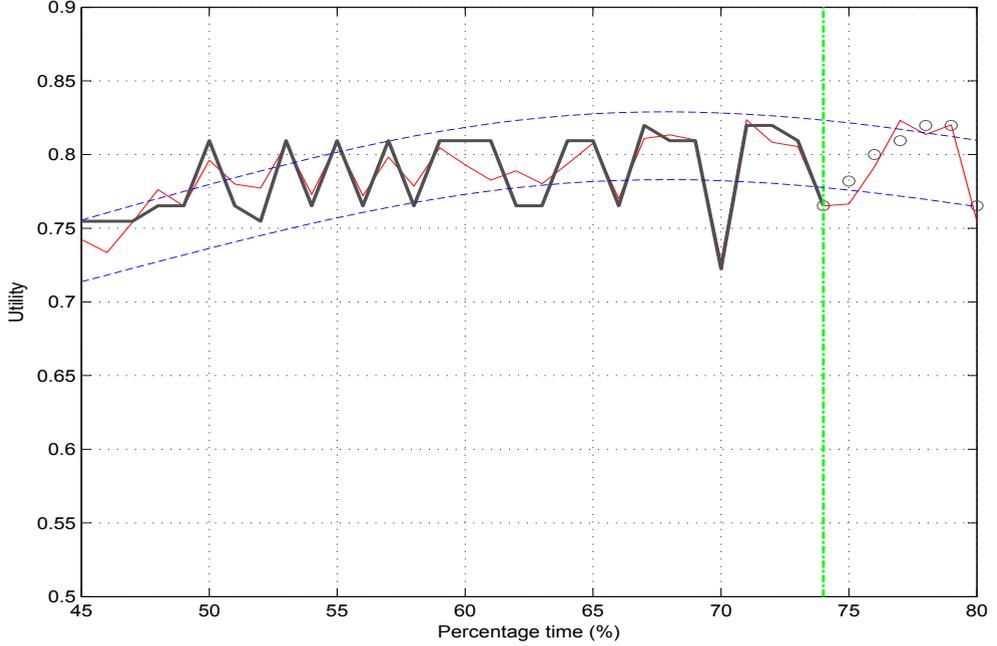


Figure 3.7: Illustrating the prediction power of our model. The original time series χ , represented by the thick solid line, is received from negotiation with agent Agent_K2 in domain *Camera*. The prediction is depicted by the thin solid line, and the two dash lines show the estimated upper and lower bounds of χ . The vertical thick dash-dot line indicates the time point at which EMAR calculated the prediction, and the circles right to this line are the utilities actually received in the subsequent negotiation phase.

residue becomes less than the predetermined value of substantial consequence. Overall, c_1 contains the signal at a fine-grained time scale and subsequent IMFs include information at increasingly longer time periods (i.e., lower frequencies). Eventually, the data series $\chi(t)$ can be expressed by

$$\chi(t) = \sum_{i=1}^n c_i + r_n \quad (3.13)$$

where n is the total decomposition layer (i.e., the number of IMFs), c_i is the i -th IMF component and r_n is the final residue (which represents the main trend of the data series). With that, EMAR is able to achieve a decomposition of the data into n empirical modes and one residue (refer to line 7). The IMFs contained in each

frequency band are independent and nearly orthogonal to each other (with all having zero means) and they change with variation of the data series $\chi(t)$, whilst the residue part captures the central tendency. An example can be found in Figure 3.6, which shows χ and resulting IMFs along with the residue part. Observation can clearly tell that the frequency characteristic of these IMFs are becoming increasingly lower, leading to the last term, i.e. r_3 , just remaining the core tendency of the original signal.

In the next stage, ARMA is used to predict all resulting components, and then ensemble them to forecast opponent behaviors (shown in line 8). ARMA is a common regression analysis model widely used in many fields, with the formal expression as follows:

$$(1 + \sum_{i=1}^p \phi_i L^i) X_t = (1 + \sum_{i=1}^q \theta_i L^i) \epsilon_i \quad (3.14)$$

where L is the lag operator, the ϕ_i are parameters for the p -order autoregressive term, the θ are parameters for the moving average term with q order, and ϵ is a parameter capturing white noise. The parameter p and q are then chosen by Akaike information criterion (AIC). AIC is a measure of the relative goodness of fit of a statistical model. For a through discussion of it, the interested reader is advised to refer to [48].

Equation 3.14 is applied with appropriate parameters for each component extracted via EMD (i.e. c_i and the residue) for the purpose of making accurate prediction, and then EMAR ensembles them to predict the future counter-offers of the opponent. Figure 3.7 exemplifies this methodology, depicting the prediction power of our model with a lead time of 6 intervals. Further details of usage are given in Section 3.2.2.

3.2.2 Adaptive Decision-making

EMAR adjusts the concession rate on the basis of the generated opponent model. Thereby a dynamic conservative expectation $R(t)$ is used to avoid “irrational concession” caused by inaccurate or, more importantly, over-pessimistic predictions. This makes sense in the case of negotiation opponents that are “sophisticated and tough” and aim at avoiding (or maximally delaying) concession making in bargaining: in this case prediction can lead to a misleading, very low expectation about the utility offered by the opponent and this, in turn, could result in an adverse concession behavior. Furthermore, using global prediction could make this situation even worse. (This phenomenon is also considered in Section 3.2.3.)

$R(t)$ guarantees the desired minimum utility at each step, yielding values which are taken as the lower bound of the agent’s expected utilities. For the purpose of adaptation to complex negotiation sessions, $R(t)$ requires two parameters e_{\min} and

u_{res} . They are both periodically updated depending on the forecast of the opponent concession (see line 10). e_{min} is defined as the minimum expectation of the compromise suggested by the opponent. Specifically, e_{min} is set to the maximum value of $\psi^{\text{low}}(t)$, which is the estimated lower bound of the predicted χ given by the central trend. Formally:

$$\psi^{\text{low}}(t) = \omega(t)(e(r_{[0,t_l]}) - \sigma(r_{[0,t_l]})) \quad (3.15)$$

where ω is the predicted main tendency of χ , $r_{[0,t_l]}$ is the series including the ratio between ω over χ within $[0,t_l]$, the operator e is the expected value and σ the standard deviation.

Having obtained ψ^{low} , e_{min} can be defined as follows:

$$e_{\text{min}} = \begin{cases} \vartheta & \text{if } \vartheta > \text{Max}(\psi^{\text{low}}(t_c + \xi)) \\ \text{Max}(\psi^{\text{low}}(t_c + \xi)) & \text{otherwise} \end{cases} \quad (3.16)$$

where $\text{Max}(x)$ returns the maximum value of input vector x and ϑ is the given reservation value. Because counter-offers with utilities indicated by ψ^{low} have already been received or can be expected in the incoming opponent moves, that is to say, our proposals around those utilities are highly possible to be accepted by the other party, using the maximum value assures an increase of the agent's potential profit at low risk of a failure.

The variable u_{res} is the dynamic reservation utility specifying the lowest expectation of the eventual benefit at the time point t_l . Formally this is captured by:

$$u_{\text{res}} = \begin{cases} \vartheta & \text{if } \vartheta > \text{Max}(\psi^{\text{low}}(t_l)) \\ \frac{1}{2}(\text{Max}(\psi^{\text{low}}(t_l)) + \vartheta) & \text{otherwise} \end{cases} \quad (3.17)$$

Because the final negotiation outcome (failure or agreement) is more sensitive to u_{res} than e_{min} , EMAR adopts a cautious and conservative way to specify it, where only $\psi^{\text{low}}(t_l)$ is considered.

Based on the above specifications, the dynamic conservative expectation function, in principle, should concede over time and dynamic reservation utility (u_{res}), whereas it is proportional to the minimum expectation (e_{min}) and the discounting factor (note a small value of discounting factor means larger time pressure). Thus $R(t)$ can be characterized as a ‘‘dynamic conservative expectation function which carefully suggests utilities’’. Equation below shows the way as used in EMAR to illustrate how it

works.

$$R(t) = \delta^\eta \cos\left(\frac{1-\delta}{1.1}t^\lambda\right)(1-t^{1/\beta})(\text{getMaxU}(P) - u_{\text{res}}) + \frac{e_{\text{min}} - u_{\text{res}}}{2}(1-t^{5\delta}) + e_{\text{min}} \quad (3.18)$$

where β and λ are concession factors affecting the concession rate, $\text{getMaxU}(P)$ is the function specifying the maximum utility dependent on a given preference P , δ is the discounting factor, and η is the risk factor which reflects the agent's optimal expectation about the maximum utility it can achieve. Here, the cosine function is used because it is a monotone decreasing function over $[0,1]$ with the codomain being between 0.5 and 1, and the discounting factor is multiplied by 5 to avoid quick concession in the early stage.

The subsequent process is then to decide the target utility EMAR expects to gain, represented by line 12. The ensemble of all predicted components provide useful information about the opponent behavior in the lead time. This is essential because the observation of ω (and its estimated bound ψ) only gives the ambiguous area where opponent would make a compromise (rather than how the move might look like). Let the predicted utility series be $E(t)$, given as follows:

$$E(t) = \sum_{i=1}^n f_i(c_i(t), \xi) + f_{n+1}(r_n(t), \xi) \quad (3.19)$$

where $f_i(x, y)$ is the corresponding prediction model for components c_i (the IMFs) and r_n (the residue). The first parameter corresponds to the input data, the second is the lead time for prediction.

Assume that the future expectation we have obtained from $E(t)$ is optimistic (i.e., there exists an interval $\{T|T \neq \emptyset, T \subseteq [t_c, t_s]\}$), that is,

$$E(t) \geq R(t), \quad t \in T \quad (3.20)$$

where t_s is the end point of the predicated series and $t_s \leq t_{max}$. In the present case the time \hat{t} at which the maximal expectation \hat{u} is reached is set as follows:

$$\hat{t} = \operatorname{argmax}_{t \in T} E(t) \quad (3.21)$$

Moreover, in this case \hat{u} is defined as

$$\hat{u} = E(\hat{t}) \quad (3.22)$$

On the other hand, now assume that the estimated opponent concession is below

Table 3.3: Overview of application domains

Domain	Issues	Size	Opposition	Discounting factor	
Travel	7	188,160	medium	1.0(U1)	0.4(D1)
Itex vs Cypress	4	180	strong	1.0(U2)	0.5(D2)
SuperMarket	6	98,784	strong	1.0(U3)	0.5(D3)
England vs Zimbabwe	5	576	medium	1.0(U4)	0.6(D4)
Energy	8	390,625	strong	1.0(U5)	0.6(D5)
NiceOrDie	1	3	strong	1.0(U6)	0.6(D6)
Amsterdam party	6	3,024	medium	1.0(U7)	0.7(D7)
Grocecy	6	1,600	weak	1.0(U8)	0.7(D8)
Camera	5	3,600	weak	1.0(U9)	0.89(D9)

the agent’s expectations (according to $R(t)$), that is, there exists no such time interval T as in the “optimistic case”. Then it is necessary to define the probability of accepting the best possible utility that can be achieved under this pessimistic expectation. This probability should be inversely proportional to the minimum adjusted difference between $D(R(t_\nu), t_\nu)$ and $D(E(t_\nu), t_\nu)$, and the discounting factor, which is formulated here as:

$$\varphi = 1 - \frac{D(R(t_\nu), t_\nu) - D(E(t_\nu), t_\nu)}{\rho \cdot \sqrt{1 - \delta D(\text{getMaxU}(P)\delta^\eta, t_\nu)}}, \quad t_\nu \in [t_c, t_s] \quad (3.23)$$

where ρ indicates the acceptance tolerance for the pessimistic forecast and t_ν is given by

$$t_\nu = \operatorname{argmin}_{t \in [t_c, t_s]} (|D(E, t) - D(R, t)|) \quad (3.24)$$

Depending on the probability φ the best possible outcome in the “pessimistic” scenario is chosen as the target utility. The rationale behind it is that if the agent rejects the “locally optimal” counter-offer (which is not too negative in accordance with ρ), it probably gives up the opportunity to reach a fairly good agreement. In the acceptance case, \hat{u} and \hat{t} are defined as $E(t_\nu)$ and t_ν , respectively. Otherwise, \hat{u} is defined as -1, meaning it does not take an effect, and $R(t_c)$ is used to set the target utility u' . When the agent expects to achieve a better outcome (see Equation 3.20), it chooses the optimal estimated utility \hat{u} as its target utility (see Equations 3.21 and 3.22).

Obviously, it is not rational and smart to concede immediately to \hat{u} when $u_l \geq \hat{u}$, nor it is appropriate for an agent to shift to \hat{u} without delay if $u_l < \hat{u}$ (especially because the predication may be not very accurate). To deal with this, EMAR simply concedes linearly. More precisely, the concession rate is dynamically adjusted in order to be able to “grasp” every chance to maximize profit. Overall, u' is calculated as

Table 3.4: Average raw score of every agent over non-discounting and discounting domains.

Agent	Non-discounting domains		Discounting domains	
	Mean	Standard deviation	Mean	Standard deviation
EMAR	0.801	0.0038	0.579	0.0026
Gahboninho	0.776	0.0085	0.481	0.0081
HardHeaded	0.744	0.0138	0.523	0.0063
Agent_K2	0.622	0.0083	0.505	0.0061
IAMhaggler2011	0.582	0.0044	0.524	0.0027
BRAMAgent	0.582	0.0078	0.506	0.0069

follows:

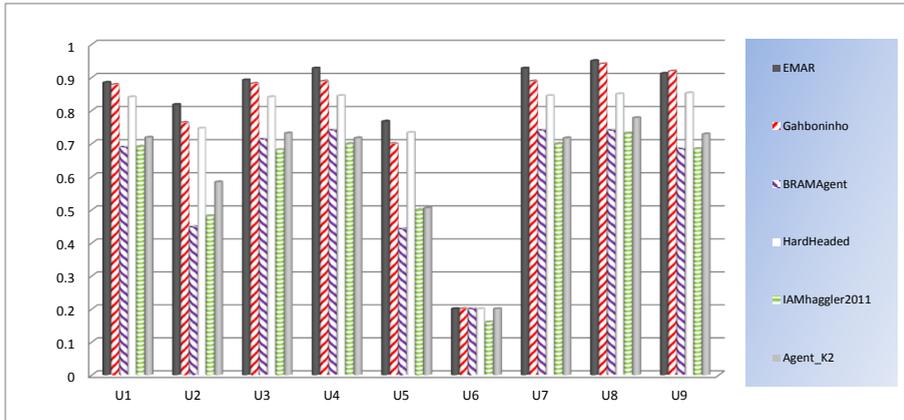
$$u' = \begin{cases} R(t_c) & \text{if } \hat{u} = -1 \\ \hat{u} + (u_l - \hat{u}) \frac{t_c - \hat{t}}{t_l - \hat{t}} & \text{otherwise} \end{cases} \quad (3.25)$$

where u_l is the utility of last bid before EMAR performs prediction process at the time point t_l .

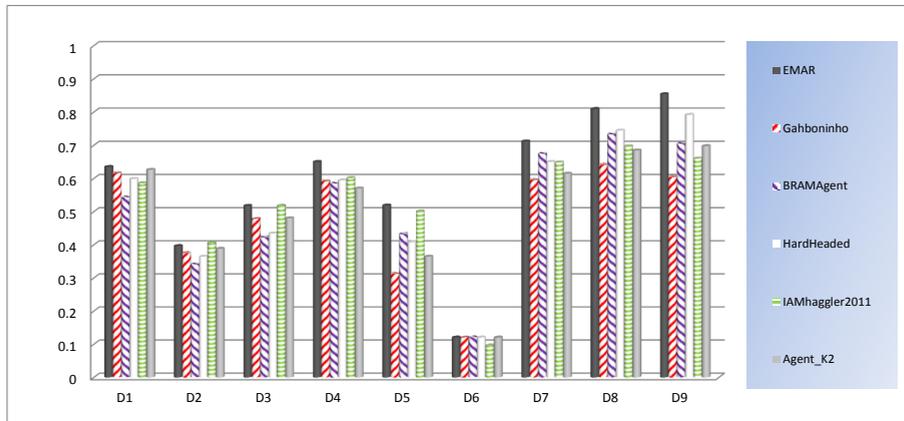
This stage corresponds to lines 13 to 18 in Algorithm 3. When the expected utility u' has been determined, the agent needs to examine whether the utility of the latest counter-offer $U(O_{\text{opp}})$ is better than u' or whether it has already proposed this offer in the earlier negotiation process. If either of these two conditions is satisfied, the agent accepts this counter-offer and finishes the current negotiation session. Otherwise, the agent constructs a new offer which has an utility within some range around u' . It is thus reasonable that an agent selects any offer whose utility is in the range $[(1 - 0.005)u', (1 + 0.005)u']$. If no such solution can be constructed, the agent makes its latest bid again in the next round. Moreover, with respect to negotiation efficiency, if u' drops below the utility provided by the best counter offer, the agent chooses that best counter offer as its next offer. This makes much sense because this counter offer can well satisfy the expected utility of the opponent who then will be inclined to accept it.

3.2.3 Experimental Analysis

In order to evaluate the performance of EMAR, GENIUS [30] is used as the testing platform. The performance of an agent (its negotiation strategy) can be evaluated via its utility achievements in negotiation tournaments which include a possibly large number of negotiation sessions over a variety of negotiation domains. Section 3.2.3 begins with describing the overall experimental setting and then presents the experimental results in the context of tournament performance.



(a) Average raw scores of all agents in non-discounting domains



(b) Average raw scores of all agents in discounting domains

Figure 3.8: Average raw scores of all agents in the ten domains. The vertical axis represents utility and horizontal axis represents domain.

Environmental setting

EMAR is compared against the best winners (i.e., the top five agents) of ANAC 2011 instead of all ANAC 2011 agents in order to make the setting more competitive; these are HardHeaded, Gabboninho, IAMhaggler2011, BRAMAgent and Agent_K2 (descending order in ANAC 2011, and for technicalities of these agents refer to [5]). Moreover, we use nine standard domains created for ANAC as testing scenarios. Around half of these domains were used in ANAC 2010 and others were from later ANAC competitions. It makes the setting avoid advantageous bias for EMAR (note that the creators of the 2011 winners knew the ANAC 2010 domains and could opti-

Table 3.5: Overall performance over all domains. The bounds are based on the 95% confidence interval.

Agent	Normalized Score			
	Mean	Standard Deviation	Lower Bound	Upper Bound
EMAR	0.718	0.0058	0.706	0.729
HardHeaded	0.623	0.0128	0.598	0.649
Gahboninho	0.617	0.0119	0.593	0.641
Agent_K2	0.520	0.0107	0.498	0.541
IAMhaggler2011	0.507	0.0063	0.494	0.519
BRAMAgent	0.494	0.0118	0.471	0.518

mize their agents accordingly).

Additionally, to evaluate the performance in domains where agent performance are affected by time-discounting factors, we equip the domains with a discounting factor. This overall setting is balanced and fair with representing a board range of important domain features, i.e., the domains vary from low opposition to high opposition, from small outcome space to large outcome space, from low time pressure to high time pressure, etc. Furthermore, the choice of domains fully covers the testing domains used by a number of recent work (refer to [13, 28, 82]), which makes the results directly comparable. For convenience, we refer to the non-discounting domain (where the discounting factor is 1) “Travel” as U_1 , “Itex vs Cypress” as U_2 , “SuperMarket” as U_3 , “England vs Zimbabwe” as U_4 , “Energy” as U_5 , “NiceOrDie” as U_6 , “Amsterdam party” as U_7 , “Grocery” as U_8 and “Camera” as U_9 . The corresponding versions with time-dependent discounting are referred to as D_1, D_2, \dots, D_9 , respectively. The application domains are overviewed in Table 3.3 with respect to four key aspects. For each domain, we run a tournament consisting of 6 agents (i.e., the five 2011 winners and our EMAR agent) 10 times to get results with high statistical confidence, where each agent negotiates against all other agents in different roles. These roles are predefined in ANAC domains and correspond to conflictive preferences like “buyer” and “seller”. The agents do not have any information about their opponents’ strategies and they are prohibited to take advantage of knowledge they might have acquired in previous negotiation sessions about their opponents. The duration of a negotiation session is 180 seconds.

Furthermore, the experiments are also done using the top agents from ANAC 2012 as benchmarks. The 2012 competition, as the latest international negotiation competition so far, was held later than the finalization of EMAR. It is therefore of interest to compare how EMAR performs playing against those most recent and advanced agents. The results are presented next, in addition to the primary results (with the best winners of ANAC 2011).

The EMAR agent divides the overall duration of a session into 100 consecutive

intervals of 1.8 seconds each. The lead time ξ is 6, the pair concession coefficients of (β, λ) is (0.04,3) and the risk factor η is 0.2, the tolerance coefficient ρ is 0.05. These values work well in practice, but we have not intended to tweak them to stay away the issues of over-fitting and unfair competition.

Primary competition results

We show the experimental results achieved by each agent in terms of raw score (i.e. the score or utility from the experimental results without being normalized) based on non-discounting domains in Figure 3.8(a), and discounting domains in Figure 3.8(b).

As depicted in the figures, the results clearly highlights the excellent bargaining skills of EMAR. Precisely, EMAR wins in twelve domains with 17.3% above the mean score achieved in these domains by the other five competing agents. EMAR made the second place for the other two domains (i.e., U9 and D2), where the performance of EMAR for these domains is only marginally (namely, 0.82% and 2.34%) below the score achieved by the best-performing agent. Most notably and impressively, it outperforms others in the domain with the largest outcome space, U5, by 33% higher than the mean score of the ANAC 2011 agents.

Table 3.4 shows the overall raw scores of all agents averaged by discounting and non-discounting domains, respectively. Our agent, on average, is the best, both in the context of negotiations where the time-discounting pressure takes effect or not. In more detail, in the non-discounting domains, the average performance of the five opponents reaches 82.6% of ours; meanwhile EMAR leads that of others by a margin of 14% in the discounting domains. Our agent moreover experiences the smallest standard deviation among all agents. Then, to have comprehensive insight of agent negotiation capability, the overall performance is summarized in the Table 3.5. Normalization is done in the standard way, using the maximum and minimum utility obtained by all agents. According to the overall performance shown in this table, EMAR is ranked number one, with an average normalized score of 0.718. This is 15.2% above the second best agent – HardHeaded –, and 30% above the mean score of all five opponents. Moreover, the performance of EMAR is the most stable – merely 54.2% of the mean standard deviation of the other agents. EMAR is followed by HardHeaded and Gahboninho; these two agents made the first two places in ANAC 2011. Agent_K2, which is an updated version of the champion (named Agent_K) in ANAC 2010, made the fourth place. We notice that the ranking is somewhat different from the final results of ANAC 2011 for other agents. Based on the experimental results we think that this is mainly caused by the participation of EMAR, leading to changes in the relative strength among the negotiating agents. To sum up, these results show that EMAR is pretty efficient and significantly outperforms in a variety

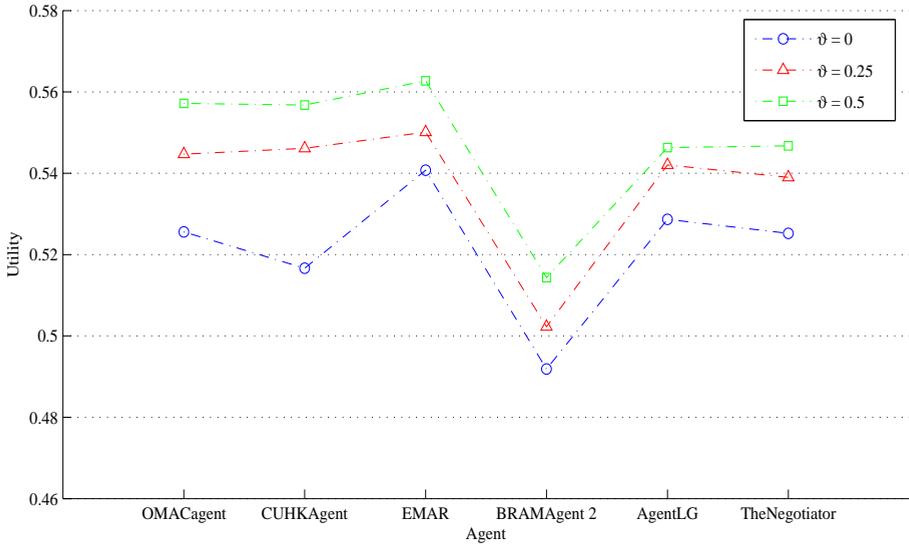


Figure 3.9: Agent performance under different reservation values when the ANAC 2012 agents are considered.

of negotiation scenarios the state-of-the-art automated negotiators (resp. negotiation strategies) currently available.

An interesting observation is that there is the noticeable gap between EMAR and IAMhaggler2011. More specifically, this agent only achieves 70.6% of the performance of EMAR in terms of normalized utility. As described in [82], similar to EMAR, IAMhaggler2011 aims at predicting an opponent’s future in order to be able to adjust its own behavior appropriately. Unlike EMAR, IAMhaggler (i) applies Gaussian process as prediction tool and (ii) adapts its concession rate on the basis of a global prediction view (i.e., on the basis of the whole preceding negotiation process). Our experimental studies suggest that a main reason for this performance gap lies in the global prediction view: this view seems to be vulnerable to “irrational concession making” induced by pessimistic predictions (see also 3.2.2). The phenomenon of irrational concession becomes increasingly apparent when IAMhaggler2011 bargains with “sophisticated and tough” opponents like HardHeaded, Gahboninho, and EMAR. For instance, when competing against these opponents in a highly competitive domain (i.e., *IteX vs Cypress*), IAMhaggler2011 only achieves a utility of 0.313 while the three opponents achieve 0.903 on average.

Table 3.6: Overall performance averaged across all domains with the best agents of ANAC 2012 included.

<i>Agent</i>	<i>Mean</i>	<i>95% confidence interval</i>	
		<i>Lower Bound</i>	<i>Upper Bound</i>
EMAR	0.551	0.541	0.561
OMACagent	0.542	0.529	0.556
CUHKAgent	0.540	0.527	0.576
AgentLG	0.539	0.526	0.552
TheNegotiator reloaded	0.537	0.523	0.551
BRAMAgent 2	0.503	0.483	0.522

Additional competition results

As an additional evaluation of the negotiation power of EMAR in demanding negotiation settings, we run a tournament with the five best agents of ANAC 2012, which include CUHKAgent, AgentLG, OMACagent, TheNegotiator reloaded, and BRAMAgent 2. Please note that we will also show later in Chapters 5 & 6 the comparison between EMAR and more advanced agents. For this tournament, we sampled the discounting factor of each domain from a uniform distribution in the interval $[0.5, 1]$ to obtain a good view of the agents' performance under different time pressure. Using such a setting that is different to the primary competition is mainly because EMAR on average performs better in the non-discounting domains (see the results given in Table 3.4), and thus the emphasis of discounting domains in the comparison with the ANAC 2012 agents would lead to more convincing and interesting results. Unlike agents submitted to ANAC 2011, the ANAC 2012 agents and EMAR are sensitive to reservation value. Therefore, we isolate the effect of reservation value on the agents' performance by averaging over the results using the same reservation value. The reservation values ϑ used here are 0, 0.25 and 0.5.

Figure 3.9 shows the mean score of each agent for each value of ϑ . Unsurprisingly the agents manage to achieve more profit as the reservation value increases. EMAR is the dominating agent regardless of the value of ϑ . The top three agents of ANAC 2012 – CUHKAgent, AgentLG and OMACagent, finish second once for respective reservation values. Table 3.6 summarizes the outcome of this tournament. As can be seen from this table, EMAR is stronger than the other agents in that it achieves the highest score among all strategies. Overall, this agent exceeds the performance averaged over the best winning strategies of ANAC 2012 by a margin of 3.6%.

4

Strategies based on Gaussian Processes

This chapter is based on: *S. Chen, H.B. Ammar, K. Tuyls and G. Weiss. Optimizing Complex Automated Negotiation using Sparse Pseudo-input Gaussian processes. In Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 707-714). 2013. IFAAMAS.*
S. Chen and G. Weiss. An Intelligent Agent for Bilateral Negotiation with Unknown Opponents in Continuous-time Domains. ACM Transactions on Autonomous and Adaptive Systems, 9(3). 2014. (In press)
S. Chen, H. B. Ammar, K. Tuyls, and G. Weiss. Transfer for Automated Negotiation. Künstliche Intelligenz (German Journal on Artificial Intelligence), Vol. 28, Issue 1, 21-27, 2014. Springer.
S. Chen and G. Weiss. An Approach to Complex Agent-based Negotiations via Effectively Modeling Unknown Opponents. (Under review)

Based upon the success of the two negotiating agents (i.e., OMAC and EMAR), this chapter makes further advancement in building opponent model and enhancing adaptive decision-making. Section 4.1 presents an improved version of OMAC, namely, OMAC*. This strategy aims at modeling opponents accurately in real-time through discrete wavelet transformation and non-linear regression with Gaussian processes. Using the approximated model the decision-making component of OMAC* adaptively adjusts its utility expectations and negotiation moves. Another negotiation strategy, called Dragon, is then described in Section 4.2 that employs sparse pseudo-input Gaussian processes (SPGPs). Specifically, Dragon enables an agent (1) to precisely model the behavior of its opponents with comparably low computational load and (2) to make decisions effectively and adaptively in complex negotiation settings. Extensive experiments are carried out for these two strategies, based on a number of negotiation scenarios and state-of-the-art negotiating agents from the 2012, 2011 and 2010 editions of ANAC.

4.1 OMAC*

This section describes OMAC*, a strategy for complex bilateral multi-issue negotiation that advances OMAC in three significant ways. First, it adopts a new learning scheme for opponent modeling that builds a regression model in real time via Gaussian processes and discrete wavelet transformation. Second, it provides an improved concession-making mechanism that takes both the agent’s estimated real reservation utility and a high-confident estimate of the forthcoming maximum opponent concession into consideration. And third, it has an enhanced response mechanism that supports an agent in selecting offers for its opponents and in determining when to withdraw from a negotiation session before the deadline is reached. Together these new features result in a considerably more efficient and adaptive negotiation strategy, as shown by the experimental results that also include a direct comparison of OMAC* and OMAC.

OMAC* is composed of two functional core components. First, an opponent-modeling component (described in Section 4.1.2), which learns a model of the opponent through a combination of discrete wavelet decomposition and non-parametric regression. Second, a decision-making component, which is responsible for adaptively making concessions (Section 4.1.3) and for appropriately responding to a counter-offer (Section 4.1.4) on the basis of the learnt opponent model. Algorithm 3 shows OMAC* at a glance, the individual steps are explained below.

4.1.1 Gaussian Processes

OMAC adopts Gaussian Processes (GPs) to learn an opponent model that does not only allow to make confident predictions but also provides a measure of the level of confidence in the predictions. GPs are an important tool in statistical modeling and are widely used to perform Bayesian nonlinear regression and classification. In the following, main aspects of GPs relevant to OMAC are overviewed; for a detailed discussion we refer to [67].

Formally, GPs are a form of nonparametric regression that perform inference directly in the functional space. Specifically, GPs define probability distributions over functions. Let $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^n$ be a data set where $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $y \in \mathbb{R}$ the output vector and n is the number of available data points. When a function is sampled from a GP, we write:

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

where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ the covariance function. $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ can fully specify a GP. A common assumption is that GPs have mean zero,

Algorithm 3 The OMAC* strategy. Let t_c be the current time, δ the time discounting factor, and t_{max} the deadline of negotiation. O_{opp} is the latest offer of the opponent and O_{own} is a new offer to be proposed by OMAC*. χ is the time series comprised of the maximum utilities over intervals. ξ is the lead time for prediction procedure and ω is the central tendency of χ obtained from DWT. $E_\delta(t)$ is the expected discounted received utility at time t . u_{res} is the estimated effective reservation utility, and e^p is the confident estimate of the maximum opponent concession with a probability of p . R is the conservative aspiration level function, u' the target utility at time t_c . As explained above, δ is the discounting factor and ϑ the default reservation value specified by the preference profile.

```

1: Require:  $t_{max}, \delta, \vartheta, \xi, p, R$ 
2: while  $t_c \leq t_{max}$  do
3:    $O_{opp} \leftarrow receiveMessage();$ 
4:    $Bids = recordBids(t_c, O_{opp});$ 
5:   if  $NewInterval(t_c)$  then
6:      $\chi \leftarrow preprocessData(t_c, Bids)$ 
7:      $\omega \leftarrow decompose(\chi);$ 
8:      $(E_\delta(t), t_l) \leftarrow predict(\omega, \chi, \xi);$ 
9:      $(u_{res}, e_{min}^p) \leftarrow updateParas(\omega, \chi, \vartheta, p, t_l);$ 
10:     $R \leftarrow (u_{res}, e^p);$ 
11:  end if
12:   $u' = getTargetU(E_\delta(t), R, \delta, t_c);$ 
13:  if  $isAcceptable(u', O_{opp}, \delta, t_c)$  then
14:     $agree(O_{opp});$ 
15:  else
16:     $checkTermination();$ 
17:     $return();$ 
18:  end if
19:   $O_{own} \leftarrow constructOffer(u');$ 
20:   $proposeNewBid(O_{own});$ 
21: end while

```

which greatly simplifies calculations without loss of generality. We also follow this view in the work.

Rasmussen and Williams [67] present a wide variety of covariance functions. In this work the *Matérn* covariance function is selected because it is robust and can be computed in real time settings:

$$k_y(\mathbf{x}, \mathbf{x}') = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu r}}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu r}}{\ell} \right) \quad (4.1)$$

with r denoting the Euclidean distance between \mathbf{x} and \mathbf{x}' . The positive parameters a and ℓ determine the amplitude and length-scale, respectively, the positive parameter ν controls the smoothness of the sample functions, $K_\nu(\cdot)$ is a modified Bessel

function [2], and $\Gamma(\cdot)$ is the Gamma function with the form $\Gamma(z) = \int_0^\infty \frac{t^{z-1}}{e^t} dt$.

As the data in GP modeling can be represented as a sample from a multivariate Gaussian distribution, we have the following joint Gaussian distribution:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_T \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{bmatrix} + \sigma_n^2 \mathbf{I}\right) \quad (4.2)$$

where \mathbf{K} is the covariance matrix, $\mathbf{K}_{ij} = k_y(\mathbf{x}_i, \mathbf{x}_j)$, σ_n the noise variance, \mathbf{I} is the identity matrix, N is the size of the training set, and T is the size of test inputs. The resulting predictive distribution is then obtained by conditioning on the observed target outputs (i.e., response variables) and is given by:

$$p(\mathbf{y}_T | \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T) \quad (4.3)$$

where $\boldsymbol{\mu}_T$ and $\boldsymbol{\Sigma}_T$ are defined as

$$\boldsymbol{\mu}_T = \mathbf{K}_{TN} [\mathbf{K}_N + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y} \quad (4.4)$$

$$\boldsymbol{\Sigma}_T = \mathbf{K}_T - \mathbf{K}_{TN} [\mathbf{K}_N + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{K}_{NT} + \sigma_n^2 \mathbf{I} \quad (4.5)$$

Finally, learning in a GP setting involves maximizing the marginal likelihood given by

$$L = \log p(\mathbf{y} | \mathbf{X}, \Theta) = -\frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi \quad (4.6)$$

where $\mathbf{y} \in \mathbb{R}^{m \times 1}$ is the vector of all collected outputs, $\mathbf{X} \in \mathbb{R}^{m \times d}$ is the matrix of the input data set, and $\mathbf{K} \in \mathbb{R}^{m \times m}$ is the covariance matrix with $|\cdot|$ representing the determinant. We briefly mention here that a desirable property of GPs is that they automatically avoid overfitting.

4.1.2 Opponent Modeling

According to OMAC*, the objective of opponent modeling is twofold: to analyze the opponent's past bidding strategy with the goal to reveal the concession trend implied by its behavior ("trend analysis"); and to predict the utilities of the opponent's forthcoming offers (for the agent) on the basis of the identified trend ("trend prediction or extrapolation"). The process of opponent modeling is captured by lines 3 to 11 in Algorithm 3. Opponent modeling is technically done through a combination of discrete wavelet transformation (trend analysis) and regression with Gaussian processes (trend prediction).

When receiving a new counter-offer O_{opp} from an opponent at the time t_c , the agent records this time stamp and the utility $U(O_{opp})$ according to the agent's own utility function (see line 4). The agent divides a negotiation session into a fixed number of intervals (denoted as ζ) of equal duration. The sequence of the highest utility at each previous interval, together with their time stamps, is taken as the basis for predicting the opponent's behavior (line 6). The motivation for using this prediction basis is twofold as discussed before. First, this degrades the computation complexity so that the agent's response time is kept low. Assume that all observed counter-offers were taken as inputs, then the agent might have to deal with several thousands of data points in every single negotiation session. This computational load would lead to a clear negative impact on the quality of negotiation in a real-time setting. Second, this reduces the risk of misinterpreting the opponent's behavior that exists in multi-issue negotiations because a small change in the utility of an opponent may result in a large utility change for the negotiator. The resulting time series consisting of the maximum utilities at each interval is referred to as χ afterwards.

To analyze the trend, the time series χ is first processed by applying discrete wavelet transformation (DWT); this is captured by line 7. The output of DWT includes an approximation and a detail component as discussed in Section 3.1.1. OMAC* focuses on the approximation part and intentionally ignores the detail part for three reasons. First, the approximation part represents the trend of the opponent concession curve and indicates how the concession of opponent will develop in the future. More importantly, it becomes more smooth (compared to the original signal, i.e. χ) to allow for accurate and robust prediction. Third, the detail parts correspond to high frequency short-term signal or random noise. Thus, these detail parts are trivial components of the original signal, and to calculate their precise predictive distribution would require a high computational effort.

Regression is then performed with Gaussian processes to forecast the opponent's future moves using the results of trend analysis. A notable advantage of Gaussian processes is that it not only provides the accurate estimation of the dependent variable(s) but also gives a measure of the level of confidence in that prediction. Since OMAC* adopts a periodical updating mechanism, it is not necessary and also not advantageous to forecast globally (i.e., from the current moment to the end of negotiation), because this is likely to generate noise that results into imprecise predictions. OMAC* limits the range of forecasting to ξ intervals to achieve efficiency and noise reduction.

4.1.3 Adaptive Concession-Making Mechanism

Based on the predictive distribution available through the learnt opponent model, OMAC* decides how to set the own target utility (see line 12 in *Algorithm 3*). One possibility is to set the target utility according to the maximum predicted utility. This is straightforward but may be ineffective. Suppose the negotiation opponents are “sophisticated and tough” and always avoid making concessions, the prediction results can then easily lead to a misleading, too low expectation about the future utility offered by the other party. This, in turn, can result in an adverse concession behavior.¹ Moreover, a global prediction approach can make this situation even worse. To deal with this “irrational concession” effect, OMAC* employs a conservative aspiration level function $R(t)$ that carefully suggests target utilities for the agent. (This negative effect is also considered in Section 3.1.4.) The R -function is based on two variables, u_{res} and e^p , where the former represents our agent’s estimated effective reservation utility and the latter represents the expectation of the maximum opponent concession. These two variables are periodically updated in dependence on the output of learnt opponent model (line 9 and 10 of *Algorithm 3*). Next, we motivate the usage of these two variables and define the R -function in detail.

Although the default reservation value of a negotiation (i.e., ϑ) is known, it is more like a “default solution” in the failure case (i.e., when no agreement is reached) rather than an indication of the actual minimum compromise the other party will make. Consider, for instance, an opponent that is cooperative in the sense that it is willing to concede more than ϑ (perhaps even in an early negotiation stage); in this case the worst possible outcome for the agent is not longer given by ϑ . The estimated effective reservation utility u_{res} is defined as follows:

$$u_{res} = \max(\omega_{max}^{low}(t_l), \vartheta) \quad (4.7)$$

where ϑ is the reservation value predefined by its own preference profile, t_l the last time point when the opponent modeling task is performed, max returns the larger value between arguments. $\omega_{max}^{low}(t)$ is the maximum value of $\omega^{low}(t)$ in $[0, t]$, which is the lower bound of ω and is defined as

$$\omega^{low}(t) = \omega(t) \cdot (\text{mean}(r_{[0,t]}) - \text{stdev}(r_{[0,t]})) \quad (4.8)$$

with ω being the main tendency of χ , $r_{[0,t]}$ the series representing the ratio between ω over χ in the interval $[0, t]$ and $stdev$ the standard deviation.

¹An opponent model that is too sensitive to opponent behavior tends towards making higher concession than necessary to reach an agreement with that opponent. Throughout this chapter this is referred to as “adverse concession behavior” and “irrational concession”.

OMAC* is sensitive to u_{res} , that means, an inappropriate setting of u_{res} would result in a negotiation failure (in the case it is too big and the agent thus tends to make no concession) or in a reduction of its potential payoff (in the case it is too small and the agent thus tends to concede more than necessary). Because ω depends on the received counter-offers, using the maximum value from its lower bound assures with low failure risk an increase of the agent's potential profit. When this value is smaller than ϑ , the agent uses ϑ instead (see Equation 4.7).

Another key factor of $R(t)$ is e^p , which aims at keeping track of the best forthcoming compromise. Toward this end, a probability parameter p is used that specifies the likelihood of the prediction (i.e., the higher p , the more confident the prediction). The definition of e^p is based on the error function that is used in the standard cumulative distribution function (CDF) of a Gaussian distribution. More precisely, the CDF is given by

$$F(x; \mu, \sigma^2) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x - \mu}{\sqrt{2}\sigma} \right) \right] \quad (4.9)$$

where erf is the error function given by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad . \quad (4.10)$$

The complement of the above cumulative distribution function represents the upper tail probability of the Gaussian distribution, and its inverse function specifies an expected value (x) of a random variable X such that X falls into the interval $(x, +\infty)$ with the given probability p . This is expressed by

$$Q(p; \mu, \sigma) = \mu - \sqrt{2}\sigma^2 \operatorname{erf}^{-1}(2p - 1) \quad . \quad (4.11)$$

In order to capture a high-confident estimate of the forthcoming maximum concession, the Q-function then takes as input a probability p , the maximum posterior mean estimate $\tilde{\mu}$, and the corresponding posterior standard deviation $\tilde{\sigma}$ in the resulting predictive distribution about χ . The probability p should be set high enough so that a strong confidence about the maximal opponent concession is ensured. Since u_{res} is the worst possible outcome, the agent takes it as the minimum value for e^p . Overall, this is captured by

$$e^p = \max(u_{res}, Q(p; \tilde{\mu}, \tilde{\sigma})) \quad . \quad (4.12)$$

The conservative aspiration level function $R(t)$ should decrease the utility expectation of an agent as time proceeds. Moreover, it should take into account the opponent behavior and the discounting effect. More precisely, $R(t)$ should be proportional to

e^p and u_{res} so that the agent is likely to benefit from the concessive behavior of an opponent. At the same time, $R(t)$ should be inversely proportional to δ because the benefit of an early agreement becomes increasingly significant as the discounting factor decreases. These requirements on $R(t)$ can be instantiated in different ways. OMAC* does this as follows:

$$R(t) = (U_{\text{max}} - u_{\text{res}})(1 - t)^{\frac{\beta}{\delta}} \left(\frac{e^p}{U_{\text{max}}} \right)^\beta + u_{\text{res}} \quad (4.13)$$

where β is the concession coefficient controlling the concession rate, U_{max} is the possible maximum utility in the scenario and δ is the discounting factor.

In addition, OMAC* uses the expected discounted received utility $E_\delta(t)$ in its decision making. This utility, which corresponds to the expectation of how much discounted profit can be received from an opponent at some future time t_* , is defined by:

$$E_\delta(t_*) = \frac{1}{C} \int_{-\infty}^{+\infty} D_\delta(u \cdot f(u; \mu_*, \sigma_*), t_*) du \quad (4.14)$$

where C is a constant called normalizing constant, f is the probability density function of Gaussian distribution, and μ_* and σ_* are the mean and standard deviation (both obtained from GPs) at t_* . Unlike the approach described in [82], which truncates the probability distribution to $[0, 1]$, OMAC* preserves the probability distribution by introducing the normalizing constant C .

OMAC* distinguishes two cases with respect to the expected discounted received utility. First, the expectation of $E_\delta(t)$ is optimistic, that is, the expected received utility is larger than what is suggested by the conservative aspiration level function. Formally, this means that there exists an interval $\{T|T \neq \emptyset, T \subseteq [t_c, t_s]\}$, where t_c is the current time slot and t_s the end point of the predicated time series, so that

$$E_\delta(t) > D^\delta(R(t), t), \quad t \in T \quad . \quad (4.15)$$

In this case the time \hat{t} at which the optimal expected utility \hat{u} is reached is set as follows:

$$\hat{t} = \operatorname{argmax}_{t \in T} E_\delta(t) \quad . \quad (4.16)$$

Moreover, \hat{u} is defined as

$$\hat{u} = E_\delta(\hat{t}) \quad . \quad (4.17)$$

Second, the expected received utility is below the suggested aspiration level. In this pessimistic case \hat{u} is defined as 0 and OMAC* abides by the target utility determined by $R(t)$. By distinguishing these two cases, OMAC* aims at “getting the most with lowest possible risk”.

Obviously, it would be not rational to concede immediately to \hat{u} when $u_l \geq \hat{u}$, where u_l is the utility of the last bid before the opponent model is updated at time t_l . Similarly, it would be not appropriate for an agent to immediately switch to \hat{u} if $u_l < \hat{u}$. Therefore, OMAC* dynamically adjusts the concession rate by setting the target utility u' as follows:

$$u' = \begin{cases} R(t_c) & \text{if } \hat{u} = 0 \\ \hat{u} + (u_l - \hat{u}) \frac{t_c - \hat{t}}{t_l - \hat{t}} & \text{otherwise} \end{cases} . \quad (4.18)$$

4.1.4 Counter-Offer Response Mechanism

After the target utility u' has been chosen, an agent has to decide how to respond to the opponent's current counter-offer (this corresponds to lines 13-19 in *Algorithm 3*). OMAC* first checks whether any of the following two conditions is fulfilled:

- the utility of the latest counter-offer $U(O_{opp})$ is not smaller than u'
- the counter-offer has been proposed by the agent itself to its opponent at some earlier point during the ongoing negotiation process.

If any of these two conditions is satisfied, the agent settles the deal and the negotiation ends (line 14).

If none of these two conditions is satisfied, then OMAC* checks whether u' falls below the best counter-offer received so far. If this is the case, then, for the reason of negotiation efficiency, this counter-offer is proposed to the opponent. Such an offer is reasonable because it tends to satisfy the expectation of the opponent. If this is not the case, then OMAC* constructs a new offer following an ϵ -greedy strategy [12, 75]. According to this strategy, a greedy offer with probability $1-\epsilon$ is chosen in order to exploit the opponent behavior, and with probability ϵ , a random offer is made (where $0 \leq \epsilon \leq 1$).² The greedy offer is chosen as follows. For a rational opponent it is reasonable to assume that the sequence of its counter-offers is in line with its decreasing satisfaction. Thus, the more frequent and earlier a value of an issue j appears in counter-offers, the more likely it is that this value contributes significantly to the opponent's overall utility. Formally, let $F(\cdot)$ be the frequency function defined by:

$$F^n(v_{jk}) = F^{n-1}(v_{jk}) + (1-t)^\varphi \cdot g(v_{jk}) \quad (4.19)$$

where the superscript of $F(\cdot)$ indicates the number of negotiation rounds, φ is the parameter reflecting the time-discounting effect, and $g(\cdot)$ is a two-valued function

²Random offers randomly select issue values (rather than utilities). No matter what kind of offers the agent chooses, their utilities are determined by the target utility u' (see Equation 4.33).

whose output is 1 if the specific issue value (i.e., v_{jk}) appears in the counter-offer and 0 otherwise. The new offer to be proposed is the one whose issue values have the maximal sum of frequencies according to the frequency function and whose utility is not worse than the current target utility. For efficiency purposes, the updating takes place for the early stage (5%) of a negotiation session. In the case of a random offer, an offer whose utility is within a narrow range around u' is randomly generated and proposed.

An important decision to be made by a negotiating agent is whether or not an ongoing negotiation should be broken off. This can make sense especially in negotiations with tough opponents if the reservation value (ϑ) is non-zero and the time-discounting effect (δ) is severe. In this situation, the agent may obtain a better payoff by aborting the tough negotiation as early as possible, namely, a slightly discounted reservation value rather than only a highly discounted outcome based on a late agreement. OMAC* uses the following probability η to decide on breaking off a negotiation:

$$\eta = \begin{cases} 0 & \text{if } \tilde{\mu} \geq \vartheta \\ (\vartheta - \tilde{\mu})(1 - \delta) & \text{otherwise} \end{cases} \quad (4.20)$$

where $\tilde{\mu}$ is the maximum mean value of the gained prediction, ϑ is the reservation value, and δ is the time-discounting factor. According to this definition, η is proportional to ϑ but inversely affected by the maximum concession prediction and δ . The rationale behind this is that a high reservation tends to make a significant negotiation success less likely, while a small discounting factor (implying high time pressure) reduces the payoff quickly. OMAC* handles breaking-off rather conservative: before really breaking off, the opponent's forthcoming counter-offers are analyzed for a certain period of time (5% of the overall negotiation time in the experiments reported below), and break-off eventually happens if none of these counter-offers is better (i.e., concedes more) than best counter-offer received so far.

4.1.5 Empirical Evaluations

Domain descriptions

We conducted a variety of experiments with domains of different complexity, where complexity is characterized by two key factors: competitiveness and domain size. Competitiveness represents the minimum distance from all of the points in the outcome space of a domain to the point leading to a complete satisfaction for both sides (note that such an ideal solution may not be always available). As competitiveness increases, it thus becomes more and more difficult to reach an agreement that meets both sides' interests. The domain size refers to the number of possible agreements or

Table 4.1: Overview of application domains for OMAC*.

Group	Domain name	Year	Issues	Domain size	Competitiveness
I	IS BT Acquisition	2011	5	384	0.117
I	Music Collection	2012	6	4,320	0.150
I	Laptop	2011	3	27	0.160
I	Phone	2012	5	1,600	0.188
II	Amsterdam party	2011	6	3,024	0.223
II	Barbecue	2012	5	1,440	0.238
II	Flight booking	2012	3	36	0.281
II	Airport site selection	2012	3	420	0.285
III	Itex vs Cypress	2010	4	180	0.431
III	Barter	2012	3	80	0.492
III	Fifty fifty	2012	1	11	0.707
III	NiceOrDie	2011	1	3	0.840
IV	ADG	2011	6	15,625	0.092
IV	SuperMarket	2012	6	98,784	0.347
IV	Travel	2010	7	188,160	0.230
IV	Energy	2012	8	390,625	0.525

the scale of the outcome space of a domain. The larger the domain size is, the more important is the efficiency of an agent’s negotiation approach because only a possibly very small fraction of the outcome space can be explored under time constraints. The domains we used were all chosen from the available ANAC domains. We group the domains into four groups: Groups I, II and III contain domains of low, medium and high competitiveness, respectively, and Group IV contains domains having a large outcome space. Specifically, Group I contains the domains *IS BT Acquisition*, *Music Collection*, and *Laptop*, *Phone*; Group II contains the domains *Amsterdam party*, *Barbecue*, *Flight booking*, and *Airport selection*; Group III consists of the domains *Itex vs Cypress*, *Barter*, *Fifty fifty*, *NiceOrDie*; and Group IV contains the domains *ADG*, *SuperMarket*, *Travel*, and *Energy*. All domains used in the experiments are overviewed in Table 4.1. For descriptions of these domains the readers are referred to [7, 26, 52].

Furthermore, for assessing the effect of the discounting factor and the reservation value on the performance of the strategies (or agents), different values for these two parameters are considered. More precisely, we conducted experiments with three discounting factor parameters (i.e., $\delta = \{0.5, 0.75, 1.0\}$) and three reservation value parameters (i.e., $\vartheta = \{0, 0.25, 0.5\}$), which resulted in nine (3×3) different scenarios for each domain.

Table 4.2: Overview of benchmark agents for evaluating OMAC*.

<i>Agent</i>	<i>Affiliation</i>	<i>Achievement</i>	
		<i>ranking</i>	<i>competition</i>
CUHKAgent	Chinese University of Hong Kong	1st	ANAC 2012
AgentLG	Bar-Ilan University	2nd	ANAC 2012
OMAC	Maastricht University	3rd	ANAC 2012
HardHeaded	Delft University of Technology	1st	ANAC 2011
Gahboninho	Bar Ilan University	2nd	ANAC 2011
IAMhaggler2011	University of Southampton	3rd	ANAC 2011
Agent_K	Nagoya Institute of Technology	1st	ANAC 2010
Nozomi	Nagoya Institute of Technology	2nd	ANAC 2010
Yushu	University of Massachusetts Amherst	3rd	ANAC 2010

As benchmark agents for the experimental evaluation of OMAC* we used the three best-performing agents of each of the 2010, 2011 and 2012 ANAC competitions. An overview of these agents, which together form a highly competitive negotiation setting, is given in Table 4.2. For details on the benchmark agents, we refer to [1, 7, 13, 28, 82].

Basic tournament and OMAC* setting

The empirical evaluation is done with GENIUS, which is the official platform used for the international ANAC competition. It allows to compare agents (representing different negotiation strategies) across a variety of application domains under real-time constraints. For each scenario of each domain, we run a tournament consisting of ten agents (including OMAC* and the nine competitors) ten times to get results with statistical confidence. In each tournament each agent repeats negotiation against the same opponent four times, where they exchange both their negotiation roles (i.e., buyer and seller role) and the order in which they start with bidding. No information about the opponents' strategies or other private information is available to any of the agents, and none of the agents can take advantage of previous encounters with their opponents (which is assured by the GENIUS platform). The maximum time for every negotiation session is 180 seconds as it is the default setting in GENIUS and ANAC competitions. When there is no agreement reached at the end of a session, then the disagreement solution applies, which means that each agent receives its own reservation value.

In our experiments OMAC*-agent performed effectively and very robust for a broad range of parameter settings. Table 4.3 shows a concrete parameter setting used in the experiments reported in this article.

Table 4.3: Overview of primary parameter settings for OMAC*.

<i>Parameter</i>	<i>Description</i>	<i>Value</i>	<i>Comment</i>
λ	decomposition level	4	no significant performance differences for more layers (i.e., values ≥ 4)
ξ	lead time	15	10% of the maximum interval, too large values decrease prediction accuracy
p	probability used for opponent concession prediction	0.9	the probability should be high enough to ensure a strong confidence about the prediction results
β	concession coefficient	0.1	the higher the value the more cooperative the agent becomes
ϵ	probability of random offers	0.5	equal chances for exploration and exploitation
φ	time-discounting coefficient	1.5	the higher the value the less important counter-offers later on are taken as

Evaluating effectiveness of opponent models

In this subsection, we compare the proposed opponent modeling scheme with other important methods that also aims at learning the opponent’s strategy by means of predicting future concession. The opponent modeling component of OMAC* is benchmarked against two main competitors, EMD+ARMA and Gaussian Processes (GPs), which are employed by EMAR [15] and IAMhaggler2011 [82], respectively. These models are applied to predict the utilities of future offers proposed by negotiation partners in all domains given in Table 4.1 with two different time-constraint scenarios: negotiations with a short negotiation deadline (i.e., 60 seconds), and negotiations with the standard ANAC deadline (i.e., 180 seconds). In this way, the performance w.r.t. small and large numbers of offer exchanges can be both assessed. Moreover, the benefit of combining the two modules (i.e., DWT and GPs) can be verified.

The results are shown in Table 4.4, where root mean square errors (RMSE) under short and standard time-constraints are listed for each model in each domain. As can be seen from this table, the performance of models tends to increase with negotiation time. More precisely, there was on average a difference of 14.4% between them. This indicates that a training of the models with more samples improves their performance in terms of prediction accuracy. Furthermore, the results show that the opponent modeling component of OMAC*-agent is also robust for short-time-period negotiations, obtaining an average RMSE of 3.53%. Moreover, OMAC*-agent outperforms others with a much higher accuracy rate in both cases. Specifically, it managed to achieve lower RMSE: around 80% (in the case of short time-constraints) and 58% (in the case of regular time-constraints) of the mean RMSE of others. With respect to regular negotiation deadlines, GPs+DWT was significantly better than the other approaches (using Welch’s t test). Overall, these results show that the new learning

Table 4.4: The RMS errors averaged over the three opponents on each domain. Bold means the value significantly better than GPs (95% confidence in each case based on Welch's t test).

<i>Opponent model</i>	<i>IS BT Acquisition</i>		<i>Music Collection</i>		<i>Laptop</i>		<i>Phone</i>	
	<i>Short</i>	<i>Regular</i>	<i>Short</i>	<i>Regular</i>	<i>Short</i>	<i>Regular</i>	<i>Short</i>	<i>Regular</i>
GPs	1.15	1.46	4.1	1.7	2.52	2.51	8.11	6.67
GPs+DWT	1.64	0.85	3.38	0.99	2.49	1.45	5.66	3.72
EMD+ARMA	4.12	3.15	5.52	4.0	3.36	3.04	6.40	6.44
	<i>Amsterdam party</i>		<i>Barbecue</i>		<i>Flight booking</i>		<i>Airport selection</i>	
GPs	4.14	6.74	7.51	6.59	5.94	6.67	3.93	3.72
GPs+DWT	4.21	3.12	6.22	4.07	5.41	3.59	3.37	1.98
EMD+ARMA	5.03	4.67	6.57	6.13	5.58	5.36	4.74	4.61
	<i>Hex vs Cypress</i>		<i>Barter</i>		<i>Fifty fifty</i>		<i>Nice or die</i>	
GPs	5.05	3.81	3.66	3.03	1.53	1.99	0.01	1.76
GPs+DWT	4.37	2.38	3.10	1.83	1.39	1.52	0.15	1.71
EMD+ARMA	3.77	3.34	3.62	3.08	1.05	1.41	1.63	2.1
	<i>ADG</i>		<i>SuperMarket</i>		<i>Travel</i>		<i>Energy</i>	
GPs	5.95	5.23	6.05	6.31	7.71	7.25	4.98	4.07
GPs+DWT	3.48	3.17	3.05	1.71	3.97	3.24	4.62	2.95
EMD+ARMA	5.24	4.18	4.24	3.92	4.47	3.83	4.03	3.80

scheme – DWT+GPs – outperforms GPs as well as EMD+ARMA.

Performance in different levels of competitiveness

Based on the domains from Group I to III, Figure 4.1 shows the agents' performance under low, medium and high competitiveness. For each domain the influence of the discounting factor and the reservation value are taken into consideration by using the resulting nine possible scenarios (as described above). As can be expected, all agents managed to increase their profit as competitiveness decreases. OMAC* was the winner in all three groups, where the distance to the other agents grows with the level of competitiveness. OMAC made the 2nd place in Groups II and III, and CHUHKAgent made the 2nd place in Group I. The performance of IAMhaggler2011 dropped dramatically as the competitiveness gets stronger.

Table 4.5 summarizes the results for Groups I, II and III. Overall OMAC*-agent was the most successful agent, which was above the average performance of other negotiators by up to a 19% in the most competitive Group III. According to our analysis the main reason for this is the ability of OMAC* to estimate with high precision the future concession an opponent will make. Due to this estimate, an agent using OMAC* concedes less especially in highly competitive domains. In less competitive domains (Groups I and II), where it is more likely that win-win solutions exist and thus agreements can be found with less compromise, this ability of OMAC* tends to have a lower impact. As can be also seen from the table, OMAC* is a clear

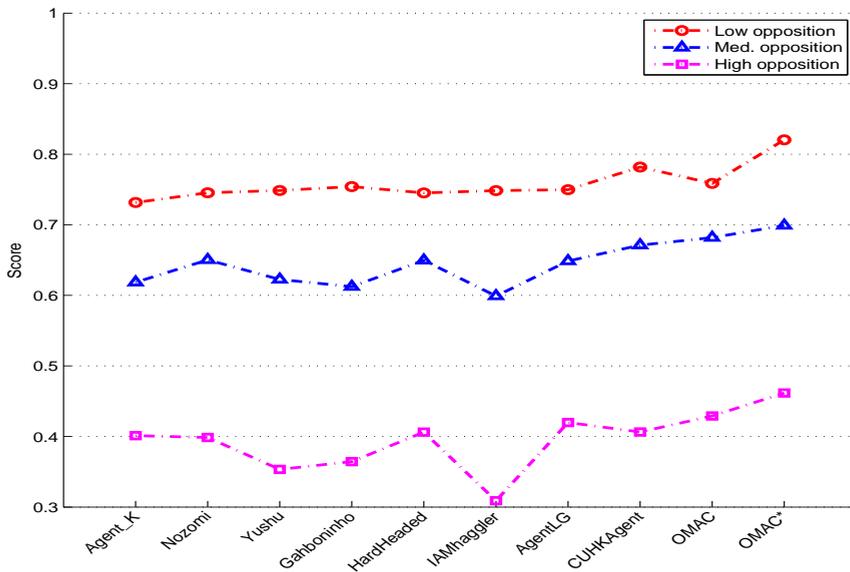


Figure 4.1: Scores of agents under different levels of opposition.

Table 4.5: Performance summary of each opposition level, ordered by the agents' overall ranking (see Table 4.8).

<i>Agent</i>	<i>Low opposition</i> (G I)	<i>Medium opposition</i> (G II)	<i>High opposition</i> (G III)
OMAC*	0.820	0.699	0.462
OMAC	0.759	0.682	0.429
CUHKAgent	0.782	0.671	0.406
AgentLG	0.750	0.649	0.420
Nozomi	0.745	0.650	0.399
HardHeaded	0.745	0.650	0.406
Agent_K	0.732	0.619	0.401
Gahboninho	0.754	0.612	0.364
IAMhaggler2011	0.748	0.599	0.309
Yushu	0.749	0.622	0.353

improvement of OMAC, which is the second-best agent. OMAC achieved scores of about 94% of OMAC*.

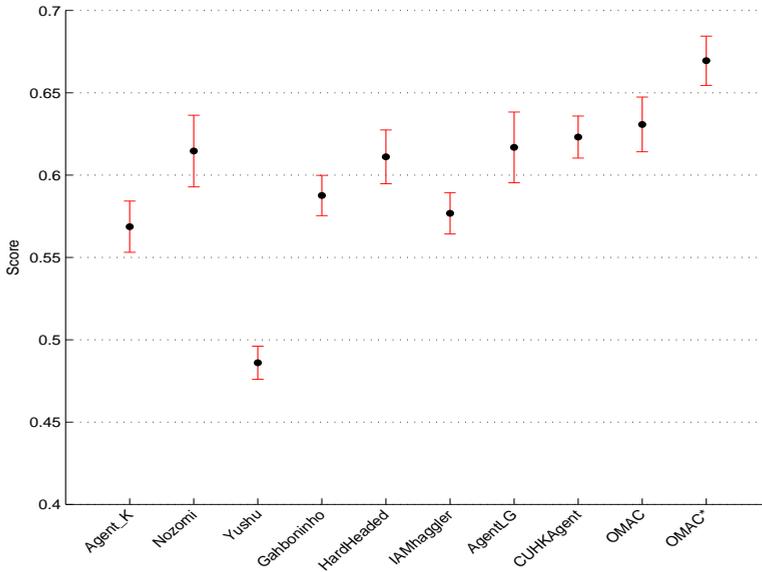


Figure 4.2: Performance of agents in the largest outcome space domains (Group IV). The score of each agent is represented by the bold dot. Error bars are indicated by vertical lines.

Performance in the largest outcome space domains

Group IV contains the four largest ANAC domains (with sizes ranging from 15,625 to 390,625) and has an average domain size of more than 170,000. The results for this group are shown in Figure 4.2. As done for the other groups (see above), the results are averaged over all scenarios of the included domains in order to cover a sufficiently broad range of values for the discounting factor and the reservation value. As can be seen from the table, OMAC* again outperformed all other competitors. OMAC and CUHKAgent finished second and third, respectively. OMAC* achieved a score of 0.669, which is 13% higher than the mean score of the opponents. As these results also show, there was an increase in the performance of the best agents of ANAC 2010, ANAC 2011, ANAC 2012. Specifically, the 2010, 2011 and 2012 agents on average obtained a score of 83%, 88% and 93% of OMAC*. OMAC performed better than the other ANAC agents, but remained 6% below the score achieved by OMAC*. Overall, the results for Group IV confirm the suitability of OMAC* for very large domains.

Table 4.6: Scores of agents averaged over all scenarios, grouped by the discounting factor and ordered by their overall ranking.

<i>Agent</i>	<i>Score</i>		
	$\delta = 0.50$	$\delta = 0.75$	$\delta = 1.00$
OMAC*	0.553	0.640	0.795
OMAC	0.506	0.613	0.757
CUHKAgent	0.504	0.615	0.743
AgentLG	0.506	0.601	0.718
Nozomi	0.502	0.598	0.706
HardHeaded	0.489	0.577	0.743
Agent_K	0.473	0.572	0.694
Gahboninho	0.497	0.518	0.724
IAMhaggler2011	0.511	0.560	0.604
Yushu	0.451	0.564	0.643

Evaluation of the impact of the discounting factor

As the discounting factor decreases, the payoff of the participants is increasingly affected over time. It is therefore interesting to investigate the performance of the agents (respectively their negotiation strategies) for different time-discounting levels. For that purpose we partition all available scenarios into three classes ($\delta = \{0.5, 0.75, 1.0\}$) according to their discounting factor. This comparison is presented in Table 4.6. As this table shows, an increase of δ (hence a decrease of the time pressure) results in an increase of the scores achieved by the agents. A comparison with the non-discounting case ($\delta = 1$) reveals that the agents' mean score dropped by 30% to 0.499 for $\delta = 0.5$ and by 18% to 0.586 for $\delta = 0.75$. Each of the three agents IAMhaggler2011, CUHKAgent and OMAC finished second in one of the three classes. OMAC* performed best in all three classes, with a performance that was 12.1% above the average performance of the others for $\delta = 0.5$, 10.3% for $\delta = 0.75$, and 13% for $\delta = 1$. Interestingly, the smallest difference between OMAC* and its opponents occurred for the medium time-discounting factor ($\delta = 0.75$). Our analysis indicates that this was the case because most opponents are optimized for a medium discounting level. Overall, these results show the ability of OMAC* to adapt effectively to different time-discounting levels.

Evaluation of the impact of the reservation value

For the purpose of better understanding the impact of the reservation value, we divide all scenarios according to the used reservation value ($\vartheta = 0, 0.25, \text{ and } 0.5$) into three classes. Table 4.7 shows the performance results of the agents achieved

Table 4.7: Scores of agents averaged over all scenarios, grouped by the reservation value and ordered by their overall ranking.

<i>Agent</i>	<i>Score</i>		
	$\vartheta = 0.00$	$\vartheta = 0.25$	$\vartheta = 0.50$
OMAC*	0.656	0.657	0.675
OMAC	0.609	0.619	0.647
CUHKAgent	0.604	0.615	0.643
AgentLG	0.585	0.607	0.634
Nozomi	0.583	0.599	0.624
HardHeaded	0.581	0.602	0.626
Agent_K	0.568	0.576	0.595
Gahboninho	0.568	0.575	0.596
IAMhaggler2011	0.553	0.557	0.565
Yushu	0.553	0.547	0.558

for each of these three classes. A general observation from these results is that the agents achieved higher scores for higher reservation values (only one agent, Yushu, performs somewhat worse for $\vartheta = 0.25$). Importantly, OMAC*-agent performed best in all three classes, and OMAC was the second best agent (achieving an average score of about 6% below OMAC*). The advantage of OMAC* over the others decreased gradually with increasing reservation values. Specifically, the largest difference (12%) was achieved for $\vartheta = 0$, whereas the difference was 10.4% and 9.6% for $\vartheta = 0.25$ and $\vartheta = 0.5$, respectively. The reason for it is that for higher values of ϑ , the other agents tend to finish a negotiation with the disagreement solution if no received proposals' payoff is better than their reservation value.³

Performance Summary

The overall performance of the agents is summarized in Table 4.8, where the normalized mean score and standard deviation are given. Normalization is done in the standard way, using the maximum and minimum utility obtained by all agents. In addition, to calculate the rank of each agent, Welch's t test was used to check for statistically significant differences between the agents' ANAC scores (also see [52]). More precisely, we computed this for every single pair of agents in order to determine with 95% confidence which agents defeat a specific agent, and which agents are beaten by that agent. (For instance, in this setting an agent that is beaten by one agent and beats seven agents is considered to have rank 2-3, that is, with 95% certainty the rank of this agent lies between 2 and 3.) The best overall performance was achieved by OMAC*-agent, with a noticeable distance of 20% above the average overall perfor-

³Note that an agent does not know the reservation value of its opponent.

Table 4.8: Overall performance of all agents across all tournaments in descending order.

<i>Agent</i>	<i>Ranking</i>	<i>Normalized score</i>	<i>Standard deviation</i>
OMAC*	1	0.667	0.010
OMAC	2-3	0.603	0.007
CUHKAgent	2-3	0.601	0.005
AgentLG	4	0.581	0.009
Nozomi	5-6	0.570	0.009
HardHeaded	5-6	0.566	0.008
Agent_K	7	0.535	0.008
Gahboninho	8	0.526	0.007
IAMhaggler2011	9	0.511	0.005
Yushu	10	0.502	0.006

mance (normalized score) of the other agents. Moreover, the performance of OMAC* was 10% above that of OMAC, which achieved the second highest score. The difference between OMAC and CHUKAgent was not significant and therefore both made the 2-3 place. Given these results, OMAC*-agent clearly outperformed all considered state-of-the-art agents in a variety of scenarios. It also performed much better than its predecessor OMAC, especially in domains with a high time-discounting factor and domains with a low competitiveness.

Interestingly, IAMhaggler2011 [82], which also employs GPs, achieved merely 77% of the scores of OMAC*. We looked into this and found that there are two main reasons causing this performance gap: IAMhaggler2011 adjusts the concession rate according to the maximum predicted utility and the corresponding time, and the prediction of an opponent’s future moves is done in a “global” way, that is, on the basis of the whole preceding negotiation process. This kind of adaptive behavior makes IAMhaggler2011 vulnerable to “irrational concession” induced by pessimistic predictions (see Subsection 4.1.3 where it is explained how OMAC* avoids this problem). The phenomenon of irrational concession becomes increasingly apparent when IAMhaggler2011 bargains in no-time-pressure scenarios with “tough” opponents. For instance, when competing against the top three agents listed in Table 4.8 in the *Amsterdam party* domain, IAMhaggler2011 obtained an average score of only 0.533, which was 55% of the mean score of those three agents.

Having compared OMAC* against the best overall performance agents (i.e., the ANAC winners), it is also of interest whether OMAC* can outperform the winner of each domain (note that the winner of ANAC is not necessarily the winner of each domain). Therefore, in the following experiments OMAC* competes in each domain

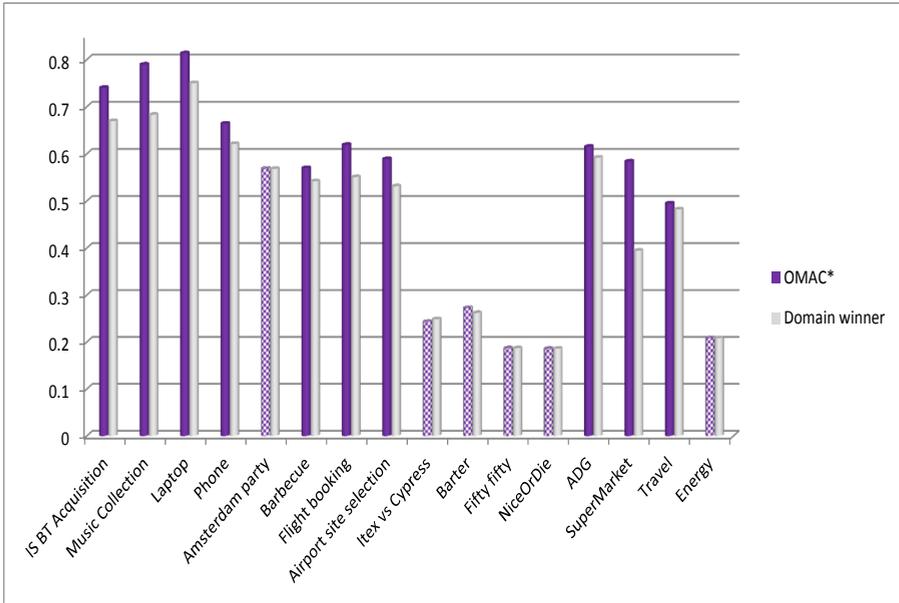


Figure 4.3: Comparing our agent against the winner of each domain. Insignificant differences are indicated by bars marked with dots.

against the domain winner. The results are given in Figure 4.3. The scoring bar of OMAC* in a domain is marked with dots if the difference between the two agents in that domain is not significant (again using Welch’s t test with 95% confidence). As can be seen, in most domains OMAC* achieved higher scores than the domain winner: the performance difference is significant in 10 domains. It was only beaten in the IteX vs. Cypress domain, but the difference was not significant.

4.2 Dragon

The overall Dragon strategy is shown in Algorithm 4. Dragon consists of three functional components, which are essential and vital for the agent to operate successfully. Firstly, the opponent-modeling component is described. It adopts a non-parametric and computationally efficient regression technique in order to approximate the opponent’s model. This allows the agent to have more accurate estimates that are used to predict the future behavior of the opponent. After having learned the opponent’s model, the concession-making component determines the optimal concession behavior using a novel adaptive decision-making strategy that automatically avoids the problem of “irrational concession”. Finally, the third and last stage of Dragon (i.e.,

Algorithm 4 The Dragon approach. Let t_c be the current time, δ the time discounting factor, and t_{max} the deadline of negotiation. O_{opp} is the latest offer of the opponent and O_{own} is a new offer to be proposed by Dragon. χ is the time series including the maximum utilities over intervals. ϱ is the lead time for prediction (i.e., the time between the release of the prediction and the end of the period being predicted), E_δ denotes the discounted expected utility of incoming counter-offers. E_{low} is the lowest expectation to negotiation, ρ is the compromise point (i.e., the time at which the agent starts to concede), and R is the dynamic conservative expectation function that carefully suggests target utilities to avoid irrational concession. u'_c is the target utility at time t_c .

```

1: Require:  $R, \delta, \varrho, t_{max}$ 
2: while  $t_c \leq t_{max}$  do
3:    $O_{opp} \leftarrow receiveMessage$ ;
4:    $recordOffers(t_c, O_{opp})$ ;
5:   if  $TimeToUpdate(t_c)$  then
6:      $\chi \leftarrow preprocessData(t_c)$ 
7:      $E_\delta \leftarrow Predict(\chi, \varrho)$ ;
8:      $(E_{low}, \rho) \leftarrow updateParas(t_c)$ ;
9:      $R \leftarrow (E_{low}, \rho)$ ;
10:  end if
11:   $u'_c = getTargetUtility(t_c, E_\delta, \delta, R)$ ;
12:  if  $isAcceptable(u'_c, O_{opp}, t_c, \delta)$  then
13:     $accept(O_{opp})$ ;
14:  else
15:     $checkTermination()$ ;
16:     $O_{own} \leftarrow constructOffer(u'_c)$ ;
17:     $proposeNewBid(O_{own})$ ;
18:  end if
19: end while

```

the responding component) responds to those counter-offers and determines the time point at which the negotiation session terminates. Next, each of the above components is detailed.

4.2.1 Sparse Pseudo-input Gaussian Processes

In the field of machine learning, Gaussian Processes (GPs) are one of the well-known, non-linear, non-parametric regression techniques. These models have been successfully applied in negotiation settings by [82]. Although GPs are a powerful form of function approximators, they suffer from computational problems once dealing with large data sets. GPs present a good candidate for opponent modeling as long as the computational complexity is reduced. To address this accuracy-computation dilemma, Dragon proposes a novel learning scheme based on Sparse Pseudo-inputs Gaussian Processes (SPGPs). These models are able to achieve similar modeling accuracy to

GPs but with much less computational effort.

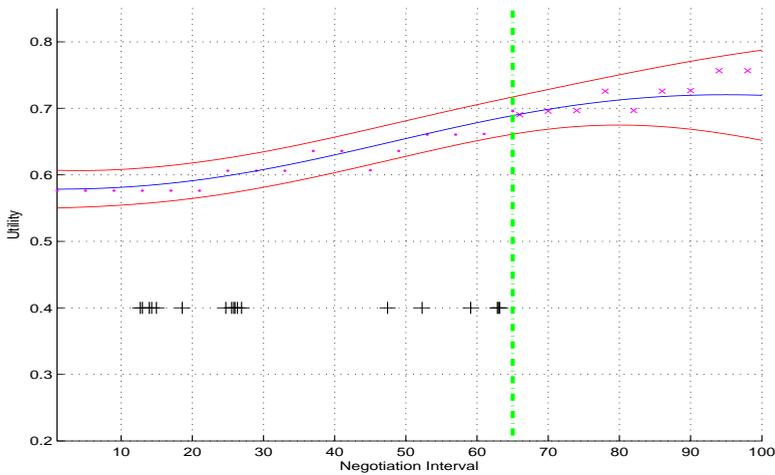
As introduced in Section 4.1.1, Gaussian Processes (GPs) are a form of non-parametric regression techniques. However, to fit the hyperparameters that best suit the available data set, we need to maximize the marginal likelihood function of Equation 4.6 with respect to Θ the vector of all hyperparameters. Typically, this maximization requires the computation of the derivatives of Equation 4.6 with respect to Θ . These derivatives are then used in a gradient-based algorithm to perform the updates. Namely, the update is performed using the following equations,

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \log p(\mathbf{y}|\mathbf{X}, \Theta) &= \frac{1}{2} \mathbf{y}^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_j} \mathbf{K}^{-1} \mathbf{y} - \frac{1}{2} \text{tr} \left(\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_j} \right) \\ &= \frac{1}{2} \text{tr} \left((\boldsymbol{\alpha} \boldsymbol{\alpha}^T - \mathbf{K}^{-1}) \frac{\partial \mathbf{K}}{\partial \theta_j} \right) \text{ with } \boldsymbol{\alpha} = \mathbf{K}^{-1} \mathbf{y} \end{aligned} \quad (4.21)$$

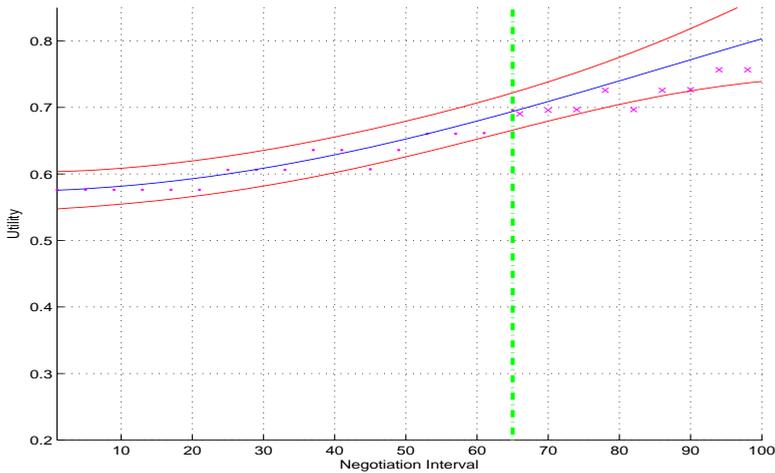
The problem with GPs is that maximizing Equation 4.6 is computationally expensive due to the inversion of the covariance matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ where n is the number of data points. The update in each step of the gradient-based optimization algorithm incurs the inversion cost of $\mathcal{O}(n^3)$. Since the covariance matrix is parameterized by the hyperparameters Θ , this inversion needs to be computed at each step of the gradient-based algorithm as the values of Θ are updated.

It is for this specific reason that we employ a fast and more efficient learning technique (i.e., Sparse Pseudo-input Gaussian Processes (SPGPs)). The most interesting feature of SPGPs is that these approximators are capable of attaining very close accuracy in both learning and prediction to normal GPs with only a fraction of the computation cost. The main reason is that learning is parameterized by a small number of pseudo-inputs that are automatically fitted depending on the variation of the sought function. This property makes them extremely suitable to the negotiation domain where a complex and low cost function approximation framework is highly demanded. The technicalities of SPGPs are described next.

As mentioned previously, normal GPs are computationally expensive to learn especially in an online setting. SPGPs aim at reducing the complexity of GPs in both learning and prediction. The idea is to parametrize the regression model with the so-called pseudo-inputs. The location of these inputs is iteratively fitted by maximizing a new kind of marginal likelihood. Interestingly, using only a small amount of pseudo-inputs, SPGPs are capable of attaining very similar fitting and prediction results to normal GPs. To clarify, the idea is to parametrize the model by $M \ll n$ pseudo-input points, while still preserving the full Bayesian framework. This leads to the parametrization of the covariance function by the location of $M \ll \ll n$ pseudo-inputs. These are then fitted in addition to the hyperparameters in order to maximize



(a) Illustration of the predictive power of SPGPs on a toy experiment.



(b) Illustration of the predictive power of GPs.

Figure 4.4: The curves in the middle represent the mean of the approximated function while the solid lines around them represent the variance. The crosses in Fig. 4.4(a) show the locations of the pseudo-inputs and the vertical dash-dot lines in both figures denote the time at which the prediction is taking place, i.e., they separate historic and predicted data points.

the new marginal likelihood, as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \Theta) &= \int p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}})p(\bar{\mathbf{f}}|\bar{\mathbf{X}})d\bar{\mathbf{f}} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \mathbf{\Lambda} + \sigma^2\mathbf{I}), \end{aligned} \quad (4.22)$$

where, $\bar{\mathbf{X}}$ is the matrix formed by the pseudo-inputs with $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}\}_{m=1}^M$. \mathbf{K}_{NM} is the covariance matrix formed by the pseudo and the real inputs as $\mathbf{K}_{MN} = k(\bar{\mathbf{x}}_m, \mathbf{x}_n)$ with $k(.,.)$ being the covariance kernel. \mathbf{K}_M^{-1} is the inverse of the covariance matrix formed among the pseudo inputs with $\mathbf{K}_M = k(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_m)$. $\mathbf{\Lambda}$ is a diagonal matrix having the diagonal entries of $\lambda_n = k_{nn} - \mathbf{k}_n^T \mathbf{K}_M^{-1} \mathbf{k}_n$. The noise variance and the identity matrix are represented by σ and \mathbf{I} , respectively.

Results in [74] show a complexity reduction in the training cost (i.e., the cost of finding the parameters of the covariance matrix) to $\mathcal{O}(M^2N)$ and in the prediction cost (i.e., prediction on a new set of inputs) to $\mathcal{O}(M^2)$. The results further demonstrate that SPGPs can fully match normal GPs with small M (i.e., few pseudo-inputs), successfully producing very sparse solutions. A full mathematical treatment may be found elsewhere [74].

In order to understand the usefulness and powerfulness of SPGPs in the context of automated negotiation, consider a classical negotiation scenario where two agents negotiate with each other about what kinds of sports to do, what time to do, the intensity as well as the price. The task of the prediction model is to forecast the utilities an agent will receive in the future. With only $M = 20$, SPGPs were able to attain very similar results to normal GPs as shown in Figure 4.4. It is clear that both learned functions follow a very similar increasing trend. Predictions made at 65 negotiation intervals also show similar predicted values in both cases. The black crosses in Figure 4.4(a) represent the location of the fitted pseudo-inputs. It is also clear that these pseudo-inputs were mostly located in critical ranges of the function.

4.2.2 Opponent-modeling Component

Modeling the opponent's behavior is done by the first component of Dragon. It adopts the SPGPs, detailed in Section 4.2.1, in order to accurately and efficiently learn the opponent's model. This process of opponent modeling corresponds to the lines 2 to 7 in Algorithm 4. Namely, upon receiving a new proposal from the opponent at the time t_c , the agent records the time stamp t_c and the utility $U(O_{opp})$ that this bid offers according to our agent's own utility function. However, in the setting of multi-issue negotiations with the two negotiation partners having different utility functions, a small change in utility of the opponent may result in a large utility

variation for our agent. Making concession at some steps does not necessarily mean the opponent will increase its concession in the coming rounds. This tends to cause a major misinterpretation of the opponent's behavior if the agent deals with every single counter-offer. Therefore and in order to reduce that negative impact, the whole negotiation is divided into a number of equal intervals (denoted as ζ). As the utility is measured in terms of our agent's own utility function, this utility may vary significantly in a given interval. Using maximum offers can alleviate the effect of noise on the prediction model. It is worth pointing out that although minimum or average offers are also useful w.r.t. noise reduction, they may to some extent lower an agent's expectation on the opponent concession, thereby causing the effect of irrational concession; see also the next section. The maximum utilities at each interval with the corresponding time stamps, are then provided as inputs to the SPGPs. As SPGPs are more computationally efficient compared to normal GPs, the number of intervals here can be much more (by factors of hundreds) than those used in [82]. This automatically leads our agent to have more accurate predictions of the future opponent's behavior compared to that work.

After learning a suitable model, SPGPs forecast the future behavior of the opponent as shown in line 7 of Algorithm 4. Dragon keeps track of the expected discounted utility based on the predictive distribution at a new input t_* , which is given by:

$$p(u_*|t_*, \mathcal{D}, \bar{\mathbf{X}}) = \int p(u_*|t_*, \bar{\mathbf{X}}, \bar{\mathbf{f}})p(\bar{\mathbf{f}}|\mathcal{D}, \bar{\mathbf{X}})d\bar{\mathbf{f}} = \mathcal{N}(u_*|\mu_*, \sigma_*^2), \quad (4.23)$$

where

$$\begin{aligned} \mu_* &= \mathbf{k}_*^T \mathbf{Q}_M^{-1} (\mathbf{\Lambda} + \sigma^2 \mathbf{I})^{-1} \mathbf{u} \\ \sigma_*^2 &= \mathbf{K}_{**} - \mathbf{k}_*^T (\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}) \mathbf{k}_* + \sigma^2 \\ \mathbf{Q}_M &= \mathbf{K}_M + \mathbf{K}_{MN} (\mathbf{\Lambda} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{NM} \end{aligned}$$

With the given probability distribution over future utilities and the effect of the discounting factor, the expected utility $E_\delta(t_*)$ is then formulated by

$$E_\delta(t_*) = \frac{1}{C} \int_{-\infty}^{+\infty} D_\delta(u \cdot p(u; \mu_*, \sigma_*), t_*) du \quad (4.24)$$

where μ_* and σ_* are the mean and standard deviation at time t_* , δ is the discounting factor.

In contrast to the work of [82] we adopt a mathematically valid approach to preserve a probability distribution by introducing C , the normalizing constant, rather than truncating the probability distribution between $[0, 1]$. The latter way does not generate a valid probability density function anymore while ours guarantees this.

4.2.3 Concession-making Component

Using the approximated model, the concession-making component aims at setting an appropriate concession degree. To deal with uncertainty in negotiation, the concession-making component primarily takes into account two factors. The first relates to the prediction of the opponent's future compromise, while the second builds on the agent's own expectation to the negotiation outcome.

As mentioned above, Dragon makes use of SPGPs to predict the future moves of an opponent following Equation 4.24. Though successful, the prediction results are sometimes rather over-pessimistic due to Boulware behavior [23] of opponents. Such “sophisticated and tough” opponents attempt to avoid (or to postpone as far as possible) to make concessions during the bargaining process. In this case an opponent's offers can easily result in a misleading, too low expectation of an agent about the utility that the opponent will offer in the future. As a consequence, an adaptive agent can be inclined to react irrationally in the sense that it compromises either too much or too early or even both. To solve the problem of “irrational concession”, Dragon employs a dynamic conservative expectation function $R(t)$. Informally, it is a “dynamic conservative expectation function that carefully suggests utilities”. Overall, $R(t)$ is sensitive to the remaining time because an agent is under more pressure to settle a negotiation as time is vanishing. Since smaller values of the discounting factor (δ) force rational agents into reaching agreements earlier (otherwise the final payoff gets more discounted), $R(t)$ is inversely proportional to δ . $R(t)$ further takes the lowest expectation (E_{low}) as its minimum value. Formally, $R(t)$ is written as:

$$R(t) = E_{low} + \left(1 - t^{\frac{1}{(1-\rho)^\beta}}\right) (u_{max}^P - E_{low}) \cos\left(\frac{1-\delta}{\omega}\right) \quad (4.25)$$

where β is the concession factor affecting the concession rate, u_{max}^P is the maximum utility of the given preference P in a domain, ω is the weight which reflects the impact of the discounting factor to the concession degree, and ρ is the compromise point.

There are two important variables considered in $R(t)$ – the compromise point and the lowest expectation. In what follows, we motivate the need for them and detail the technicalities for defining them.

Since the design objective of the negotiation strategy/agent is to maximize its profits, it is highly demanded to exploit the opponent. But, the trade-off between exploitation and compromise, is also of major importance. To clarify, suppose that the agent never makes any concession, probably no agreement would be reached, or the opponent might even break-off somewhere within the negotiation process. Thus ρ is used to adaptively adjust the time at which Dragon should stop exploiting the opponent and rather start to compromise. To this end, the value of ρ should increase

with the increasing ratio between the number of new solutions and the total solutions proposed by the other party. This is because a high ratio of new solutions tends to be a good indicator of the opponent's cooperation. Thus, we introduce γ_t to represent the ratio of new counter-offers over the past ten intervals up to t . The observation of new counter-offers cannot guarantee the concession by the other party (e.g., these new offers could just be the offers with high utility for the opponent while low utility for our agent). Therefore, the effect of γ_t is influenced by the maximum concession till t (λ_t), leading to the following:

$$\rho = 1 - (1 - \gamma_t^{(1-\lambda_t)^{\frac{\alpha}{8}}})t \quad (4.26)$$

where α is the parameter controlling the influence of λ_t .⁴

The other variable needed to define $R(t)$ is E_{low} , which represents the lowest expectation to a negotiation session. Formally, E_{low} is defined as:

$$E_{low} = \begin{cases} \theta & \text{if } \theta \geq \max U(E_{\delta=1}(0, t_l)) \\ \max U(E_{\delta=1}(0, t_l)) & \text{otherwise} \end{cases} \quad (4.27)$$

where θ is the reservation value specified by the preference, $\max U$ returns the maximal utility from counter-offers, t_l is the last time the update was carried out.

Based on the above definitions, the decision of how to counter the opponent is made as follows. If the future expectation obtained from $E_{\delta}(t)$ is optimistic (i.e., there exists an interval $\{T|T \neq \emptyset, T \subseteq [t_c, t_s]\}$), that is:

$$E_{\delta}(t) \geq D_{\delta}(R(t), t), \quad t \in T \quad (4.28)$$

with t_s being the end point of the prediction and $t_s \leq t_{max}$. Then the time \hat{t} at which the maximum expectation \hat{u} is reached is set according to:

$$\hat{t} = \operatorname{argmax}_{t \in T} E_{\delta}(t) \quad . \quad (4.29)$$

And \hat{u} is defined as:

$$\hat{u} = E_{\delta}(\hat{t}) \quad . \quad (4.30)$$

Conversely, in the pessimistic case where the estimated opponent concession is below the agent's expectations, we define the probability of accepting the best possible utility, φ , to be inversely proportional to the minimum difference between $E_{\delta}(t)$,

⁴The maximum concession (until time point t) is the maximum compromise among all counter-offers given by the opponent until time t .

$D_\delta(R(t), t)$, and the discounting factor, as follows:

$$\varphi = 1 - \sqrt[5]{\frac{\delta \cdot (D_\delta(R(t_\nu), t_\nu) - E_\delta(t_\nu))}{D_\delta(u_{max}^P, t_\nu)}}, \quad t_\nu \in [t_c, t_s] \quad (4.31)$$

where t_ν is given by

$$t_\nu = \operatorname{argmin}_{t \in [t_c, t_s]} (|E_\delta(t) - D_\delta(R, t)|) \quad . \quad (4.32)$$

According to the probability φ , the best possible outcome in the ‘‘pessimistic’’ scenario is chosen as the target utility. The rationale here is that if the agent rejects the ‘‘locally optimal’’ counter-offer, it may lose the opportunity to reach a fairly good agreement earlier.

In the acceptance case, \hat{u} and \hat{t} are defined as $E_\delta(t_\nu)$ and t_ν , respectively. Otherwise, \hat{u} is defined as -1 , meaning it does not have an effect, and $R(t_c)$ is used for the target utility u'_c . When the agent expects to achieve a better outcome (see Equation 4.28), the optimal expected utility \hat{u} is chosen as its target utility (see Equations 4.29 and 4.30).

Obviously, conceding immediately to \hat{u} is not rational for the agent when $u_l \geq \hat{u}$ (u_l is the utility of last bid before t_l). Neither is it to shift to \hat{u} without delay if $u_l < \hat{u}$, especially because the predication may be not completely accurate. To deal with this, Dragon simply concedes linearly. More precisely, the concession rate is dynamically adjusted in order to be able to ‘‘grasp’’ every chance to maximize profit. Overall, the process to set u'_c is shown in line 11, which is calculated as follows :

$$u'_c = \begin{cases} R(t_c) & \text{if } \hat{u} = -1 \\ \hat{u} + (u_l - \hat{u}) \frac{t_c - \hat{t}}{t_l - \hat{t}} & \text{otherwise} \end{cases} \quad (4.33)$$

4.2.4 Responding Component

This is the last component of the Dragon strategy and corresponds to lines 12 – 18 of Algorithm 4. After the expected utility u'_c has been determined, the agent needs to examine one of two conditions in response to the opponent. According to the first condition the agent has to validate whether the utility of the counter-offer $U(O_{opp})$ is better than u'_c , while according to the second condition the agent has to determine whether it had already proposed this offer (i.e., the opponent’s counter-offer) earlier in the negotiation process. If either one of these two conditions is satisfied, the agent accepts it and terminates the session as shown in line 12.

On the other hand, if none of them are met, the agent proposes a new offer de-

pending on an ϵ -greedy strategy. That is to select either a greedy action (i.e., exploit) with $1-\epsilon$ probability or to select a random action with an ϵ probability, where $0 \leq \epsilon \leq 1$. The greedy action is determined based on a frequency analysis. Although simple, such a method has been successfully applied by some state-of-the-art negotiating agents, like Hardheaded and CUHKAgent (refer to [5, 28]). In the present work, Dragon considers that the opponent is rational. More precisely, Dragon assumes that the sequence of counter-offers is in line with a decreasing order of satisfaction. Thus, for a value of an issue j , the more frequent and earlier it is proposed by the negotiation partner, the more contribution it makes to the opponent's overall utility.

Formally, let $F(\cdot)$ be the frequency function defined as:

$$F^n(v_{jk}) = F^{n-1}(v_{jk}) + (1 - t)^\psi \cdot g(v_{jk}) \quad (4.34)$$

where the superscript of $F(\cdot)$ indicates the number of negotiation rounds, ψ is the parameter reflecting the discounting effect of time, and $g(\cdot)$ the two-valued function, whose return is 1 if the specific issue value (i.e., v_{jk}) appears in the counter-offer and 0 otherwise.

With a probability $1 - \epsilon$, Dragon then picks the offer whose issue values have the maximal sum of frequencies according to the frequency function. In the case of the random action (similar to way as employed in OMAC), Dragon constructs a new offer which has an utility within some range around u'_c .

One additional step is needed to cope with terminating the negotiation in advance when $\theta > 0$ and $\delta \neq 1$ because in such a case failing to reach an agreement leads to a lower payoff (note the reservation value is also discounted), which is more likely to happen in a negotiation against a tough opponent. The responding component investigates whether the maximum expectation obtained from SPGPs is larger than θ . If that is true, the agent could expect to gain a better outcome than what the disagreement solution generates. Otherwise, the agent breaks off the bargaining if the opponent does not make any compromise in the incoming 10 intervals.

4.2.5 Empirical Evaluations

The performance of Dragon is evaluated by means of the official platform of the ANAC competition – GENIUS [30]. This simulation environment helps facilitate comparing different agents across a variety of application domains under real-time constraints. This section first describes the experimental technicalities including the implementation details of Dragon (referred to as Dragon-agent). The competition results are then reported. Finally the performance of negotiating agents are then summarized.

Table 4.9: Overview of test domains, ascending order of competitiveness.

<i>Domain</i>	<i>Issues</i>	<i>Domain size</i>	<i>Competitiveness</i>
DogChoosing	5	270	0.051
Kitchen	6	15,625	0.063
Animal	5	1,152	0.110
Acquisition	5	384	0.117
Icecream	4	720	0.148
Laptop	3	27	0.160
Planes	3	27	0.165
Outfit	4	128	0.198
Camera	6	3,600	0.219
SmartPhone	6	12,000	0.237
HouseKeeping	5	384	0.281
Wholesaler	7	56,700	0.308
DefensiveCharms	3	36	0.322
Lunch	6	3,840	0.420
Coffee	3	112	0.486
Ultimatum	2	9	0.545
Fifty2013	1	11	0.707
NiceOrDie	1	3	0.840

Experimental setup

In order to assess Dragon in a highly competitive setting, we set up two tournaments where its implementation (Dragon-agent) plays against the well-performing agents of the 2010 – 2012 ANAC⁵ excluding those best ones (primary tournament), and the winners of ANACs (advanced tournament). Moreover, the negotiations are conducted in the whole set of domains created for ANAC 2013. Such a setting avoids advantageous bias as they are unknown to any participant. To capture the influence of the discounting factor δ and the reservation value ϑ on the agents' performance, different values for these two parameters are considered. Thus, the experiments are conducted for each domain with three discounting factors (i.e., $\delta = \{0.5, 0.75, 1.0\}$) and three reservation values (i.e., $\vartheta = \{0, 0.25, 0.5\}$), which result in nine (3×3) different scenarios for every domain.

All domains used in this work are over-viewed in Table 4.9, where issue refers

⁵Because of the significant change with respect to the competition rules, the agents of ANAC 2013 are allowed to use historical knowledge. Introduction of them to the experiments will cause unfairness to others, and this new feature moreover violate the requirements of complex negotiation (see Section 2.2.1).

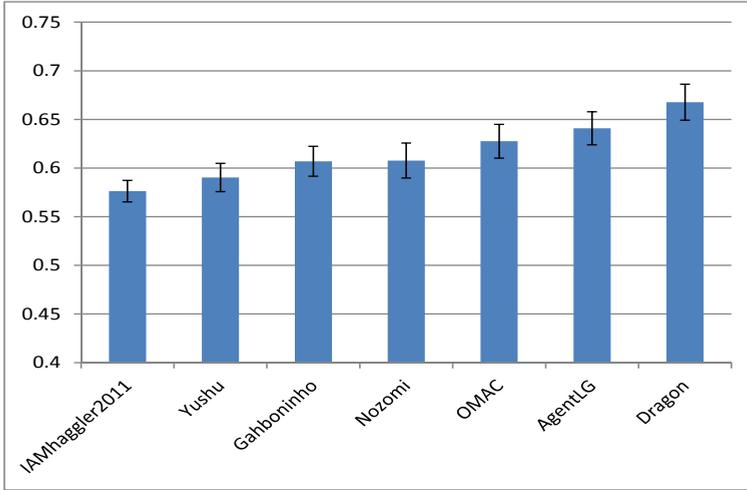


Figure 4.5: Average normalized score of negotiating agents in the primary tournament. Error bar indicates a single standard deviation.

to the number of items under negotiation, competitiveness represents the minimum distance from all of the points in the outcome space of a domain to the point leading to a complete satisfaction for both sides (note that such an ideal solution may not be always available) and the domain size means the scale of the outcome space of a domain, in other words, the number of possible agreements. For the results to be statistically significant, each scenario is repeated 20 times.

Dragon-agent simply sets ζ , which is the number of equal intervals into which the agent divides the whole negotiation, to 180. Note that ζ can be larger than the value, we use it for the purpose of convenience and robustness. The lead time ρ is limited to 25 intervals, and the parameter α for compromise point is set to 0.1. The parameters for the expectation function R are set to $\beta = 1.5$ and $\omega = 1.2$. Note that this set of parameters is chosen according to our experience without systematically optimizing it. In our experiments Dragon turned out to perform well and very robust for a quite broad range of parameters.

Results of primary tournament

The results achieved by the agents in the primary tournament are shown in Figure 4.5, where a total number of 162 scenarios are considered, resulting from 18 domains with nine different combinations of discounting factors and reservation values. As depicted in the figure, Dragon-agent demonstrated excellent performance against a variety of opponents, which was the best performing agent from the perspective of the mean

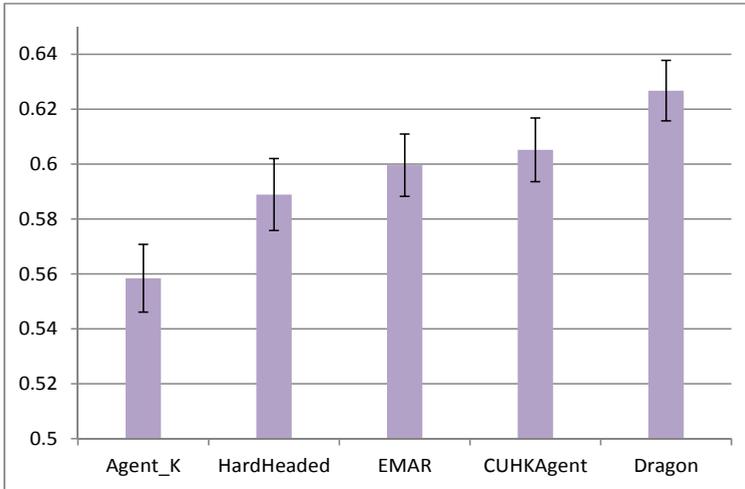


Figure 4.6: Average normalized score of negotiating agents in the advanced tournament. Error bar indicates a single standard deviation.

normalized score.⁶ More precisely, Dragon-agent achieved a score of 10% higher than the mean normalized score of opponents. AgentLG and OMAC (the second and third best agent of ANAC 2012) followed our agent and on average obtained a score of 0.634, 5% below Dragon-agent’s. Gahboninho and IAMHaggler2011, the 2nd and 3rd in ANAC 2011, presented a lower performance, around 87% of ours, and there was a similar performance difference for the agents from ANAC 2010 (i.e., Nozomi and Yushu).

Results of advanced tournament

Figure 4.6 gives the results of the advanced tournament where only the strongest opponents – the winners of each year’s competition and an additional agent EMAR [15] – are included. This setting is much more challenging than the former. Due to this fact, the performance of Dragon-agent to some extent dropped. It, however, was still ranked first among other contenders with a score of 0.627, leading the mean performance of others by a margin of 7.5%. Also, its performance experience the smallest variance. The second best agent of the advanced tournament was CUHK-Agent (the 2012 winner). EMAR lagged behind CUHKAgent with a tiny difference (within 0.005) and finished third. Then, Hardheaded (the 2011 winner), Agent.K

⁶For convenience of comparing performance across domains, normalization is adopted for the results of all scenarios and done in the standard way, using the maximum and minimum raw score obtained by all agents.

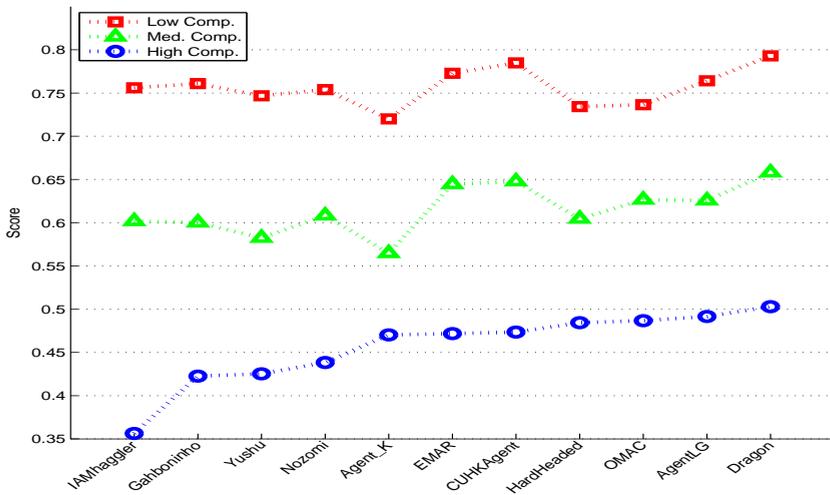


Figure 4.7: Comparing performance of the agents in different levels of competitiveness.

(the 2010 winner) took the fourth to fifth place in sequence. To sum up, Dragon-agent managed to outperform other best agents in this very competitive setting with a notable advantage.

Impact of different competitiveness

For evaluating how competitiveness affects agents' performance, the domains are classified into three groups (each with the same number of domains) to represent the different level of competitiveness as follows, low (competitiveness ≤ 0.160), medium ($0.160 < \text{competitiveness} < 0.32$), high (competitiveness ≥ 0.32). Figure 4.7 demonstrates the comparison of all agents' performance under low, medium and high domain competitiveness. The influence of the discounting factor and the reservation value has been already taken into consideration in the manner as mentioned before. Not surprisingly, all agents managed to increase their profit with a decreasing competitiveness level. Dragon-agent was the most successful agent in all three cases, where the difference to the other agents grew from 5.5% to 11.5% as the level of competitiveness increased. CUHKAgent made the 2nd place in the case of low and medium competitiveness, whereas AgentLG made the 2nd place in highly competitive domains. The performance of IAMhaggler2011 dropped most significantly as the competitiveness gets stronger. The results showed that Dragon-agent was the most effective agent regardless of the level of competitiveness.

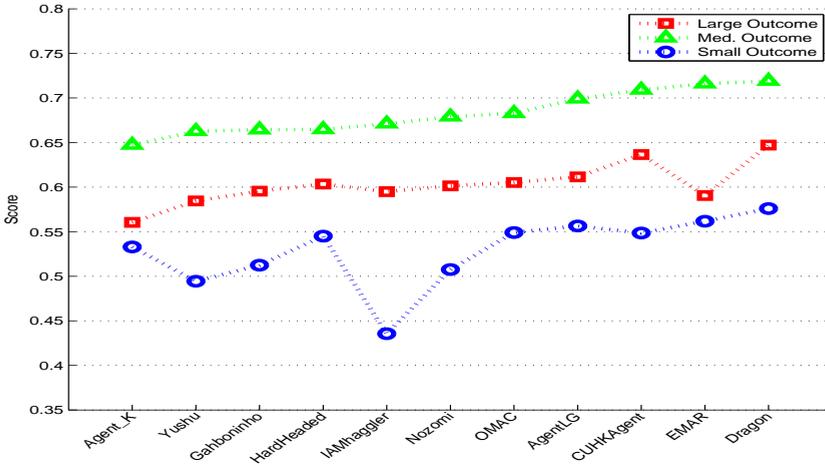


Figure 4.8: Comparing performance of the agents in different levels of domain size.

Impact of different outcome space

To assess effects of domain outcome space on agents' performance, the domains are classified into three classes to represent the different level of outcome space as follows, small (outcome space ≤ 100), medium ($100 < \text{outcome space} < 3600$), large (outcome space ≥ 3600). The results are shown in Figure 4.8, which are averaged over all scenarios of the included domains as we did in the previous subsection. As can be seen from the figure, these agents tended to achieve better negotiation results (e.g., higher scores) in medium-size domains than in small domains. Dragon-agent advanced other competitors in all three classes of outcome space domains. In large domains, CUHK-Agent and AgentLG finished in the second and third place, respectively. EMAR and CUHKAgent performed quite well in medium outcome space domains, which were only second to Dragon-agent. EMAR and AgentLG, following Dragon-agent, were ranked second and third in small domains. By considering the agents' performance in a plenty of scenarios with various classes of domain outcome space, Dragon was well suited to domains with a broad range of outcome spaces.

Performance summary

According to the overall performance as shown in Table 4.10, Dragon was the best negotiating agent considered in the experiments. The top three agents of ANAC 2012 achieved a mean score of 0.626, and the best agents of ANAC 2011 and 2010 obtained

Table 4.10: Overall performance of all agents across all scenarios in descending order.

<i>Agent</i>	<i>Normalized utility</i>	<i>95% confidence interval</i>	
		<i>Lower Bound</i>	<i>Upper Bound</i>
Dragon	0.651	0.625	0.677
CUHKAgent	0.635	0.611	0.659
EMAR	0.629	0.604	0.654
AgentLG	0.627	0.601	0.652
OMAC	0.616	0.591	0.642
HardHeaded	0.607	0.584	0.630
Nozomi	0.600	0.574	0.625
Gahboninho	0.594	0.572	0.616
Agent_K	0.585	0.561	0.608
Yushu	0.584	0.563	0.606
IAMhaggler2011	0.571	0.552	0.590

a similar mean score around 0.59, which in general presents a decreasing trend. With an average normalized score of 0.651, our agent led a margin of 8% over the mean score of these opponents. Then, CUHKAgent, EMAR, AgentLG and OMAC took the second to fifth place in sequence. In addition, as the confidence intervals of several agents overlap with Dragon’s confidence interval, Welch’s t test (with 95% confidence) was applied to the results to investigate whether the score differences are significant. The test showed that the score of Dragon was significantly better than the scores achieved by all other agents. To summarize, Dragon clearly outperformed with a considerable margin the state-of-the-art automated negotiators in a variety of application scenarios. Dragon-agent’s ability of learning opponent behavior with high precision and avoiding irrational concession may account for this success.

Moreover, the noticeable performance gap between Dragon-agent and IAMhaggler2011 is also interesting. Because both agents employ GPs or a variant of GPs to model opponents, but IAMhaggler2011 was the worst-performing one in our experiments. Unlike our agent, IAMhaggler2011: (1) applies Gaussian process as a prediction tool and (2) adapts its concession rate fully on the basis of global predictions. The empirical evaluation suggests that one reason for this performance gap lies in the global prediction view. In more detail, this view seems to be vulnerable to “irrational concession” induced by pessimistic predictions. Dragon already avoided such a behavior as explained in Section 4.2.3. The phenomenon of irrational concession becomes increasingly apparent when IAMhaggler2011 bargains with opponents in non-discounting domains where other players have no pressure to make early concession. Another reason may be due to the fact that SPGPs is more suitable than GPs

in the context of automated negotiation since SPGPs are computationally efficient so that more information can be processed to build the opponent model.

Part III

Game-theoretic Analysis of
Negotiation Strategies

5

Empirical Game-theoretical Analysis

This chapter is based on: *S. Chen and G. Weiss. An efficient automated negotiation strategy for complex environments. Engineering Applications of Artificial Intelligence 26, 10(2013), 2613-2623. 2013.*

S. Chen, H.B. Ammar, K. Tuyls and G. Weiss. Conditional Restricted Boltzmann Machines for Negotiations in Highly Competitive and Complex Domains. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI, pp.69-75). 2013. AAAI Press.

S. Chen and G. Weiss. An Approach to Complex Agent-based Negotiations via Effectively Modeling Unknown Opponents. (Under review).

The experimental analysis that we presented in the previous two chapters investigates the performance of negotiation strategies from the common mean-scoring perspective using a tournament setting. Although this analysis gives valuable insight (and thus is common in the field), it says little about the robustness of the strategies because the basic setting of the tournament is fixed, e.g., the number of participating players and strategies. For example, a strategy that achieves good performance in a tournament shows its capability of reaching reasonable agreements with a wide range of other strategies. This strategy is, however, not necessarily the best one that outperforms opponents in different tournaments composed of a different number of agents with different sets of available strategies. As a matter of fact, this analysis says nothing about the robustness of a strategy in the cases agents are allowed to switch their strategy.

For a strategy that is proven to be robust in a certain scenario, it manages to “survive” in this competitive environment in the sense at least some proportion of players stick to it, no matter how the population of players change their strategies in order to increase their own payoffs. To address robustness appropriately, a game-

theoretic analysis technique, known as empirical game theoretic analysis (EGT) [41], is applied to the tournament results. It was originally used to study the results of the Trading Agent Competition (TAC) – an international forum designed to promote and encourage high quality research into the trading agent problem. The purpose of this analysis technique is to identify pure Nash equilibria where none of the agents has an incentive to change its current strategy. In case no such equilibrium exists, we attempt to find the best reply cycle [85]. Such a cycle consists of a set of profiles (e.g., the combination of strategies chosen by players) for which a path of statistically significant single-agent deviations (whose definition is given next) exists that connect them, with no deviation leading to a profile outside of the set. These two types are both referred to as empirical equilibria, and only pure-strategy profiles are considered, in which each agent chooses a single strategy instead of a mixed one. In the following, we start by looking into important aspects of the EGT techniques. Then, we perform the EGT analysis on the results of a range of negotiation tournaments in order to investigate the robustness of strategies in various settings.

5.1 Methodology

In EGT analysis a profile/state consists of a combination of strategies used by players in the game, where some of them may use the same strategy. Strictly speaking, it is called the pure profile since players are only allowed to use a single strategy instead of choosing a strategy probabilistically. The payoff of each strategy in a profile is determined by the tournament results (see [80] for a description of this approach). A profile is represented by a node in the resulting graph. To study the behavior of agent switching strategies (or profile transition), we consider the best single-agent deviations (as done in [82]), where there is an incentive for an agent to unilaterally change its strategy in order to statistically improve its own profit, given the strategies of others are known. As more than one stable profile may be found at a given game, to evaluate the relative importance of stable profiles, we adopt the notion of basin of attraction similar to [5]. The basin of attraction refers to the portion of profile that lead to a particular state through a series of best single-agent deviations. For a best reply cycle, its basin of attraction is summed by each stable profiles inside it. The size of the basin of attraction can serve as an indicator of the likelihood of reaching that stable profile if in each initial state of the game any of the strategies is equally chosen by those agents.

We calculate the scores of our four proposed strategies and the respective top three strategies from ANAC 2012 & 2013 (i.e., **A**gent**L**G, **C**U**H**K**A**gent, **O**MAC, **F**awkes, **M**eta Agent, **T**MF Agent) averaged over all scenarios considered in tournament competitions; as a result, the payoff matrix indicating their average performance (against

Table 5.1: Strategy-pair payoff matrix, where the score pair in each entry is averaged over all domains, with the score representing the column player’s payoff (as the matrix is symmetric). The first letter (**bold**) of each strategy is used as the identifier. (Note that there is one exception that the identifier of OMAC* is **S**.)

Strategy	E	A	C	O	M	T	F	D	S
E	0.576	0.574	0.646	0.554	0.612	0.637	0.585	0.629	0.657
A	0.597	0.586	0.558	0.557	0.592	0.502	0.532	0.579	0.627
C	0.611	0.671	0.604	0.612	0.604	0.617	0.567	0.630	0.633
O	0.567	0.567	0.596	0.577	0.573	0.585	0.549	0.614	0.638
M	0.631	0.679	0.657	0.629	0.638	0.647	0.659	0.605	0.660
T	0.592	0.601	0.514	0.513	0.541	0.517	0.475	0.626	0.618
F	0.656	0.600	0.666	0.606	0.661	0.620	0.242	0.635	0.651
D	0.586	0.578	0.607	0.574	0.653	0.615	0.565	0.559	0.585
S	0.588	0.569	0.578	0.514	0.635	0.506	0.570	0.542	0.615

each strategy) can be created as shown in Table 5.1. The first letter of each strategy is used as its identifier (except that OMAC* is referred to as **S**). Because the matrix is symmetric, we only present the column strategy’s payoff. On the basis of this payoff matrix, we in the subsequent sections perform the game-theoretic analysis of repeated negotiation scenarios that is more open than those in this sections.

5.2 Underlying Two-Player Negotiation Games

In this case, we apply EGT analysis to the single negotiation case where two players are involved and each of them can freely choose among the strategies that are considered in the experiments described above. For brevity, in the following each strategy is referred to by a single letter, namely the respective bold letter in Table 4.8 (e.g., **C** stands for **CUHKAgent**). Let S be the complete strategy set, that is, $S = \{\mathbf{E}, \mathbf{O}, \mathbf{D}, \mathbf{S}, \mathbf{A}, \mathbf{C}, \mathbf{M}, \mathbf{T}, \mathbf{F}\}$. The score of a strategy in a specific profile is the payoff achieved when playing against the other strategy and averaged over all scenarios considered in the work.

The analysis results are shown in Figure 5.1. The first row of each node gives the pair of strategies of a profile; the second row shows the mean score of the strategy pair. This average score is used as a measure of the social welfare achieved by the two involved strategies, which can be interpreted as overall benefit achieved by a profile for all involved agents. The stronger strategy in each profile is marked with a color background. Each arrow indicates a statistically significant single agent-deviation to a different strategy profile. Under this EGT analysis, no pair of strategies is in equilibrium; instead, there exists a best cycle of statistically significant single-agent

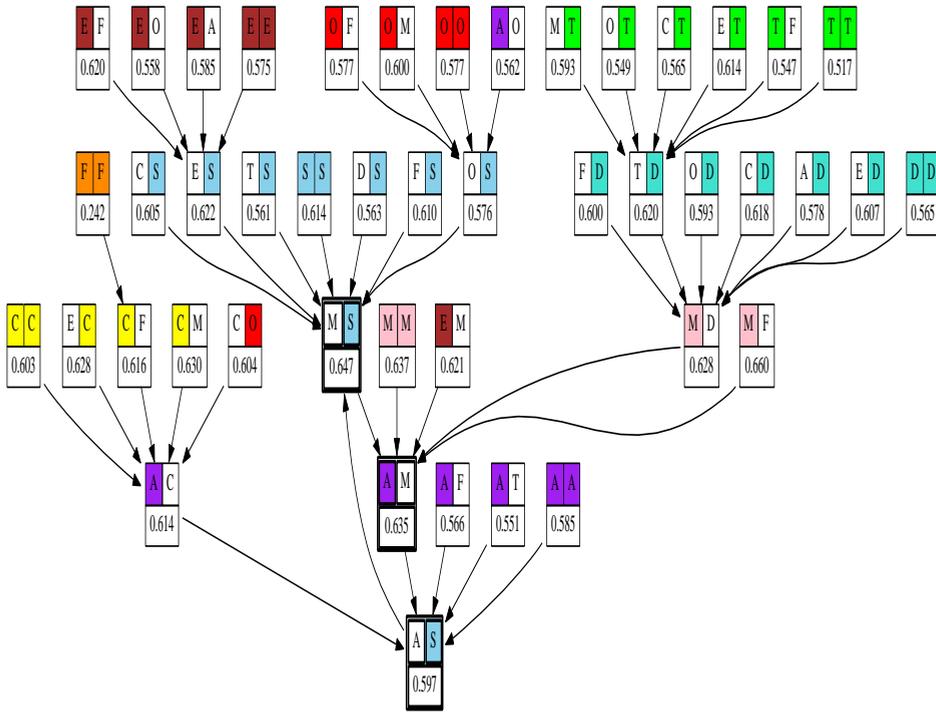


Figure 5.1: Deviation analysis of two-player encounters composed of all strategies. Each node shows a strategy profile. The first row of a node shows the involved strategies, where the strategy with the higher score is indicated by a colored background. The second row gives the average score of the strategy pair. An arrow indicates a statistically significant single-agent deviation between strategy profiles. The profiles in the best cycle are the nodes with a bold frame.

deviations. This best cycle contains three profiles, namely, $M|S$, $A|M$, $A|S$, all of which are, to highlight their importance, marked with a thicker border. For any other strategy profile not included in this cycle, there exists a path of statistically significant deviations (i.e., strategy changes) that lead to a profile within the cycle. Moreover, the highest social welfare (i.e., 0.647) is achieved by one profile in the best reply cycle ($M|S$).

To sum up, in this simple two-player negotiation encounters there are three strategies – OMAC*, AgentLG, Meta-Agent – that are robust in the sense that they are all in the empirical equilibria into which all other possible strategy combinations eventually lead (or in other words, the basin of attraction is 100%) and which are chosen by the negotiating players with equal probability (because they are symmetric). It is, however, important to see that the single encounter analysis, while useful,

says anything about the strategy robustness when the setup gets more complicated. Therefore, in the following we look into a setting composed of more players and three top strategies.

5.3 Small Negotiation Tournament Games

We now consider a six-player tournament setting where the players can choose from among a pool of three (top) strategies one that maximizes their own payoff. The first strategy set consist of three top strategies in terms of average tournament performance (i.e., Dragon, OMAC* and Meta-Agent). Besides, it is also of interest to see how the three robust strategies in two-player encounters perform. Thus, the other set we consider includes AgentLG, OMAC* and Meta-Agent.

Regarding the first strategy set, the strategy deviations are visualized in Figure 5.2, where each node represents a profile, the first row of each node lists the three strategies, and the second row shows how many players use each strategy. The strategy with the highest score is marked with a color background. In this restricted 3-strategy scenario, there exists only one equilibrium and this equilibrium contains Meta-Agent and OMAC*. It is clear that for any non-Nash equilibrium strategy profile there exists a path of statistically significant deviations that leads to this equilibrium state, that is, its basin of attraction is 100%. Moreover OMAC* is the winner of the tournament specified by the equilibrium state. The three players using the strategy Meta-Agent are not interested in switching to OMAC* (though it lags behind OMAC*), because it offers them a profit which is better than the profit that could be achieved in an OMAC* self-play setting. On the other hand, there is no incentive for the other players to switch from OMAC* to the other strategy because this would result in a decrease of their benefit.

Regarding the second strategy set, the results are more complicated as given in Figure 5.3. In addition to the equilibrium aforementioned, there exists another equilibrium where all players deviate to AgentLG. The basin of attraction is 65% and 35% for these two equilibria, respectively. This shows an interesting phenomenon – even though AgentLG is worse than OMAC* and Meta-Agent in tournament competitions, it could be the most favorable strategy in some specific scenarios in which its portion is more than a half, provided that the strategy pool is relatively small, i.e., merely having OMAC* and Meta-Agent.

This analysis of small scale negotiation games presents some insights different with the case of two-player encounters. It suggests that the introduction of a new strategy might lead to different equilibria despite the fact that the new strategy is a weak in tournament competitions. A new question, according to the results, has been arisen how players would behave given that a wider range of strategies are available.

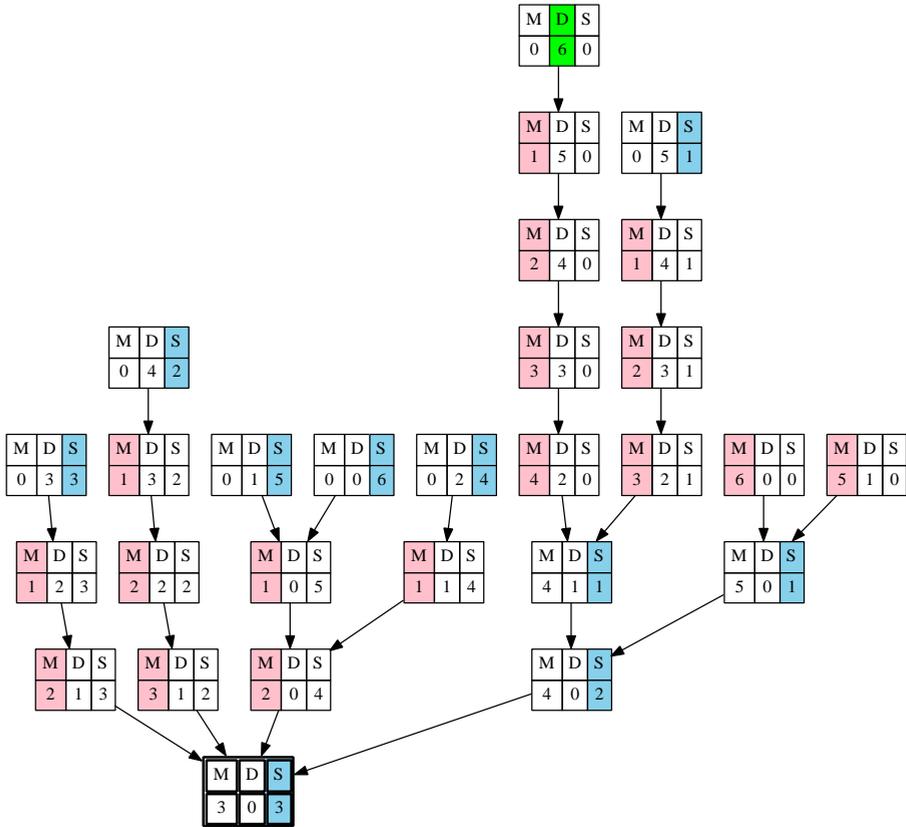


Figure 5.2: Deviation analysis of a three-player with three-strategy scenario setting. Each node represents a strategy profile, where the strategy with a color background achieves the highest score. Arrows indicate the statistically significant deviations between strategy profiles. The node with a bold frame is the only equilibrium (no outgoing arrow).

5.4 Large Negotiation Tournament Games

In this section, investigations are carried out for a further complicated case – the negotiation games that include all strategies involved, thus with nine players that freely choose any one from the nine strategies given in Table 5.1. Visualization of the complete resulting graph is not possible due to the large number of distinct nodes. More precisely, the complete graph includes $\binom{|p|+|s|-1}{|p|} = \binom{17}{9} = 24,310$ distinct nodes, where $|p|$ is the number of players and $|s|$ is the number of strategies. Therefore, we have to prune the graph so that we can concentrate on relevant features in a way

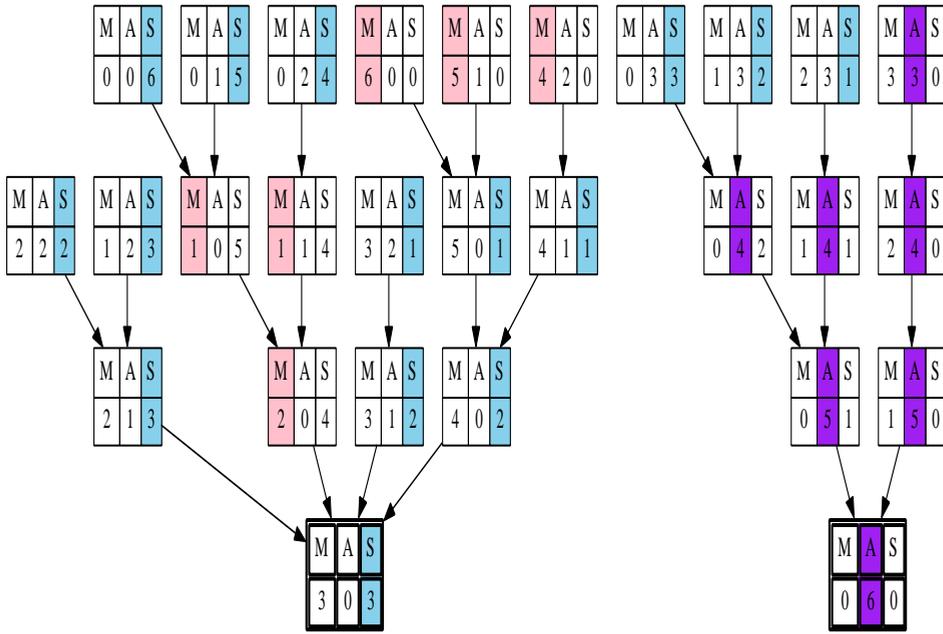


Figure 5.3: Deviation analysis of a three-player with three-strategy scenario setting. Each node represents a strategy profile, where the strategy with a color background achieves the highest score. Arrows indicate the statistically significant deviations between strategy profiles. The node with a bold frame is the only equilibrium (no outgoing arrow).

similar to [5]. In more detail, we blank all nodes in the graph except those being on a path that starts either with an initial profile where all players choose the same strategy or with an initial profile where each agent uses a different strategy and that ends with a pure Nash equilibrium. The resulting graph is shown in Figure 5.4. The first row of each node indicates the involved strategies, and the second row gives the number of players using each of the strategies. A strategy not used by any players is not displayed in order to keep the graph as compact as possible. Using this EGT analysis, it can be seen that there is only one equilibrium profile, namely the one where five players choose OMAC* and the other four are in favor of Meta-Agent; any other strategy profile eventually converges to this equilibrium state. In this more complex game, AgnetLG is no longer in any stable state (please recall the results shown in Section 5.3), which means it is not robust enough related with OMAC* and Meta-Agent. OMAC* and Meta-Agent are both in the equilibrium state, and the players with OMAC* can obtain higher payoffs than their counterparts using Meta-Agent.

towards avoiding further compromise given that the built opponent model predicts a increase in received payoff from the opponents. The key difference between them lies in the techniques they employ to build opponent models. Despite effectiveness, prediction accuracy of these two strategies suffers from the simple regression models they use.

Inheriting the framework from OMAC, OMAC* uses a more powerful regression model based on Gaussian processes to tackle the problem of prediction inaccuracy. In addition to that, its concession-deciding component is enhanced by taking into account the opponent's preferences over multiple offers and by keeping track of the appropriate termination time. As a result, OMAC* can better capture behavioral patterns of opponents and is able to find a better balance between aggression and concession in negotiation, thereby being more adaptive to the opponents and achieving a higher individual payoff (and even a better social welfare in a range of scenarios). Overall, Dragon behaves similar to OMAC*, but it has an advantage when it is about dealing with negotiation tasks in which the number of exchange offers is huge. Moreover, a simplified concession-making component is adopted to relieve computation efforts.

The empirical game-theoretic analysis of this chapter provides game-theoretic insights into the proficiency of the proposed negotiation strategies. We evaluate them against a range of top strategies in scenarios with increasing complexity, starting from simple two-player encounters to the negotiation games involved of a wide array of players. According to the results, three strategies – OMAC*, Meta-Agent and AgentLG – were reported to be robust.

Among other state-of-the-art strategies, OMAC* is particularly stable in various types of negotiation games. A main reason for this lies in its adaptive and flexible style of concession making – OMAC* always attempts to minimize its concession, but adaptively relaxes its minimization effort if the opponent behaves in such a way that disagreement becomes likely. Due to its opponent learning and decision-making scheme, OMAC* achieved high scores against opponents as well as a good performance in self-play. Moreover, the experiments demonstrate that OMAC* clearly improves over its predecessor OMAC when playing against others as well as against itself. Another impressive strategy under EGT analysis is Meta-Agent. A high self-play performance, together with a fairly good advantage over opponents, may account for its success. The third strategy (AgentLG) is stable in quite few cases – only if the strategy pool is narrow and it acquires the support of at least a half of the participants.

6

Evolutionary Game-theoretical Analysis

This chapter is based on: *S.Chen and G. Weiss. An Intelligent Agent for Bilateral Negotiation with Unknown Opponents in Continuous-time Domains. ACM Transactions on Autonomous and Adaptive Systems, 9(3). 2014. (In press)*
S. Chen and G. Weiss. Using evolutionary game theory to evaluate complex agent-based negotiation strategies. (Under review)

The EGT analysis carried out in the last chapter is based on the assumption that each player is matched against all other players; to put it differently, global interaction is assumed. In realistic multi-player systems (MAS) players, however, do not (or not need to) interact with all other players. There exist many real-life cases concerning local interaction. For example, in diplomatic negotiations on a territorial dispute it is obvious that negotiation concerns only adjacent countries rather all countries with which the disputing countries are in some relationship. In the problem of decentralized resource allocation in a wireless sensor network, each individual sensor node just need to adapt its operation in response to the status and feedback of its neighboring nodes. Moreover, using EGT analysis is not suitable for a game consisting of numerous players; otherwise the structure of the resulting graph will be extremely complicated, thus making it hard to be well studied. The negotiation games where local interaction within a spatial structure between a large number of players is considered thus have not been well studied so far. Against this background, we investigate how a population of players or individuals behave by changing their negotiation strategies in the case of local interaction. Toward this end, evolutionary game theory, more precisely spatial evolutionary game theory [43, 76], is applied to the tournament results. This allows to analyze the impact on fitness (i.e., how well an individual is adapted to a dynamic environment) of each species (strategy) competing with others locally. In this chapter, we first introduce related details of spatial evolutionary game theory. This technique

is then applied to analyze spatial negotiation game in which a population of players, with freedom to use any one from a set of negotiation strategies, interact with others locally.

6.1 Methodology

Classic evolutionary game theory is related to the theory of dynamic adaptation and learning in (infinitely) repeated games played by rational agents [76]. When a graph that specifies the location of each player and connectivity between them is further considered, the game evolves into a spatial one. Spatial evolutionary game theory [43, 76] is thus deployed to analyze the strategy dynamics in such games. In a spatial negotiation game we assume that a population of players/agents in a certain area, for the sake of higher fitness, uses a set of negotiation strategies to compete against each other. By using spatial evolutionary game theory the impact on fitness (i.e., how well a individual is adapted to a dynamic environment) of each species (strategy) competing against others locally can be well studied.

The fitness of an individual or a cell (as it is located at a certain environmental position) is determined by the average payoff of its own strategy playing against its neighbors. Take a simple case with three strategies as an example, where the center cell choosing strategy 1 meets its neighbors as shown in Figure 6.1(a). The fitness of the center cell is the average payoff of playing against three opponents using strategy 1 and three opponent using strategy 2. In other words, it has the neighbor distribution $x=(0.5,0.5,0)$. The payoff matrix of the three strategies is given by matrix A below:

$$A = \begin{bmatrix} 4 & 10 & 0 \\ 1 & 4 & 9 \\ 3 & 7 & 4 \end{bmatrix}$$

where an entry $A(i,j)$ is the payoff of strategy i against strategy j . Thus, the fitness (ρ) of the center cell is 5.0, following the equation:

$$\rho = e_i A x^T \tag{6.1}$$

where e_i denotes the i^{th} row of a unit matrix e with the size of the number of strategies and A denotes the payoff matrix. For clarity, the fitness of every cell is shown in Figure 6.1(b).

Our analysis assumes that the players in the game have the freedom to choose one of the nine negotiation strategies as in the two-player negotiation encounters. Each strategy is initialized with a equal number of players, randomly distributed over a

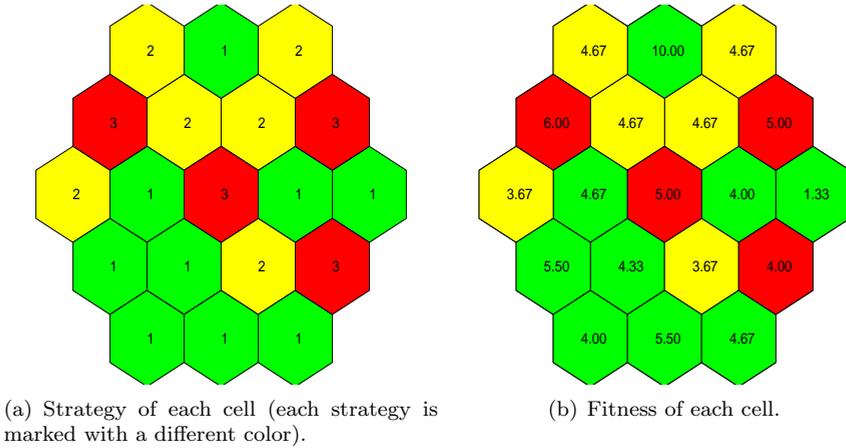


Figure 6.1: A locality example in a hexagon model.

two-dimensional hexagon lattice Λ . A cell (I) in the lattice represents a player and is occupied by one strategy and bordered with six adjacent cells, that is, each cell has six neighbors in its local scale. Calculating the fitness of each cell in the field is simultaneously performed. Each cell then imitates which one has the highest fitness in its neighborhood (including itself). In this way the natural selection process (i.e., how to update the strategy of the cell for the next generation) is well defined.

Figure 6.2 works as a toy example illustrating how a population of players switch between a number of strategies (given in Table 5.1) in such a environment. In the first generation, which is shown in Figure 6.2(a), every cell is randomly assigned with a strategy, with each strategy being indicated by a distinct color. The initial share of the strategies in the population are made equivalent to be 100 players. After the population evolves to the third generation (see Figure 6.2(b)), the two strategies (red and green) are adopted by increasingly more agents, whereas the portion of other strategies gets smaller. In the 10th generation (Figure 6.2(d)), the competitive advantage of these two strategies results in the fading away of other candidates, thus making only these two strategies co-exist in the game. As the evolution continues (see Figure 6.2(d)), the share of red strategy continues to move up and almost reaches the same level with the green strategy.

6.2 Dynamics of Three Strategies with Local Interaction

The dynamics of strategy distribution, as a topic of central interest for evolutionary game theory, can be used as a key indicator of strategy performance. In this section

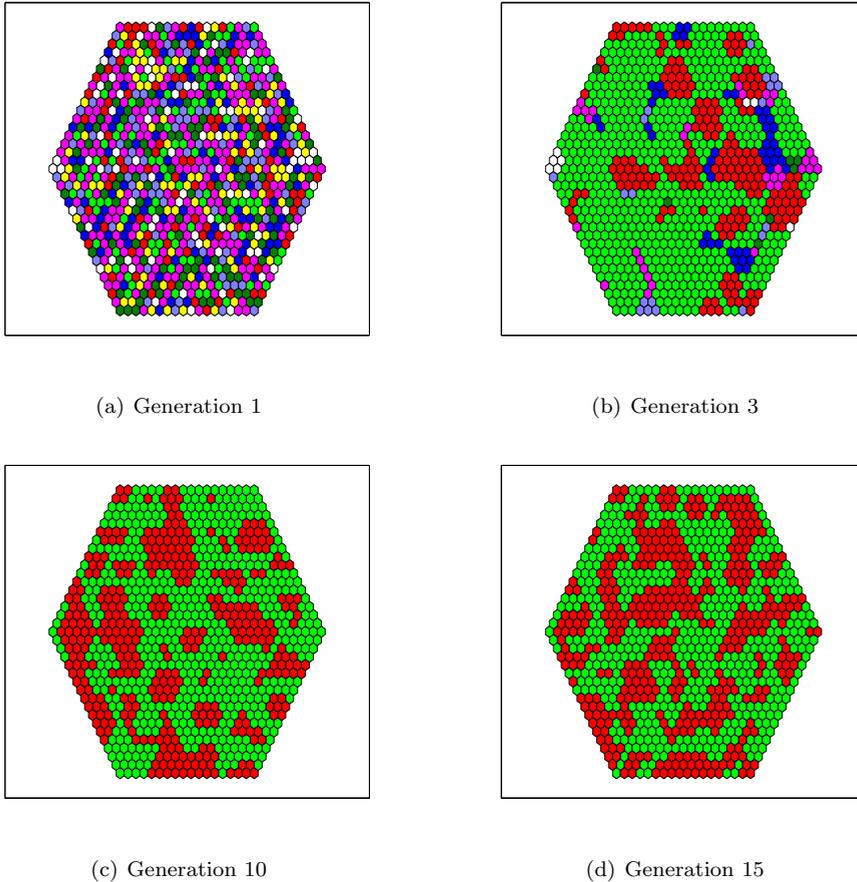


Figure 6.2: An example of strategy distribution evolution.

we start by studying how a small strategy set of three strategies evolves with respect to the population share. The strategy sets we used here are identical to the two sets in Section 5.3 such that the spatial effect on strategy performance can be clearly compared. To guarantee results with high reliability, the spatial game was simulated 10,000 times with random initialization of the location arrangement of the strategies.

Figure 6.3, regarding the first set, shows the variation of strategy proportions over generations. Meta-Agent was the leading strategy in the beginning, however its share gradually went down till generation 90. In contrast, there was a slow but steady increase in the number of players choosing OMAC*, turning out to be the most popular strategy in the end. The third strategy – Dragon was exterminated within

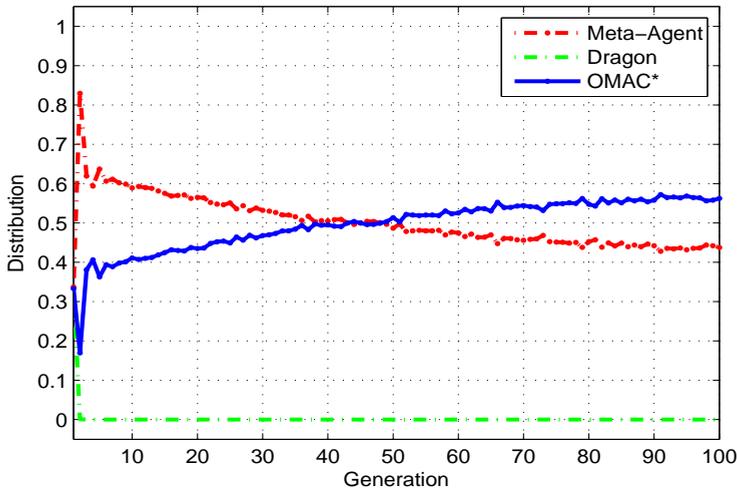


Figure 6.3: Distributions of three strategies over generations when players interact with their nearest neighbors.

one generation (as synchronized strategy update accelerates this process). After the 90th generation, the share of OMAC* and Meta-Agent was almost stable, with the former leading by a margin of 10%.

It is also interesting to investigate the results when the interaction range of a cell is extended to its next nearest-neighbors. In doing so, we would have a clearer insight on the impact of interaction range. Toward this end, the nature selection process is also modified such that a player is permitted to mimic the strategy used by the most successful player in this wider neighborhood. The results are depicted in the Figure 6.4. In contrast to the former version, the movement of population share was similar for first steps but much slower. But OMAC* could not achieve a higher proportion than Meta-Agent after sufficient length of evolution; instead there was oscillation between them in terms of population share. The result was caused by its competitive advantage over the opponents.

As regards the other set of three strategies, the outcome was similar for the two cases of interaction range. More specifically, AgentLG outperformed other two competitors in both cases of interaction between nearest neighbors (see Figure 6.5) and interaction between nearest and next-nearest neighbors (see Figure 6.6). On the other hand, OMAC* promptly reached a very high level of proportion, more than 80% of the players; meanwhile, Meta-Agent's share reduced to zero within 12 generations.

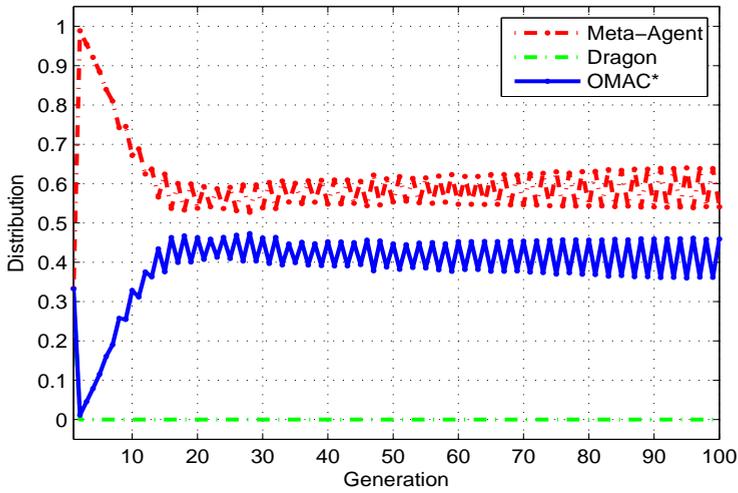


Figure 6.4: Distributions of three strategies over generations when players interact with their nearest and next-nearest neighbors.

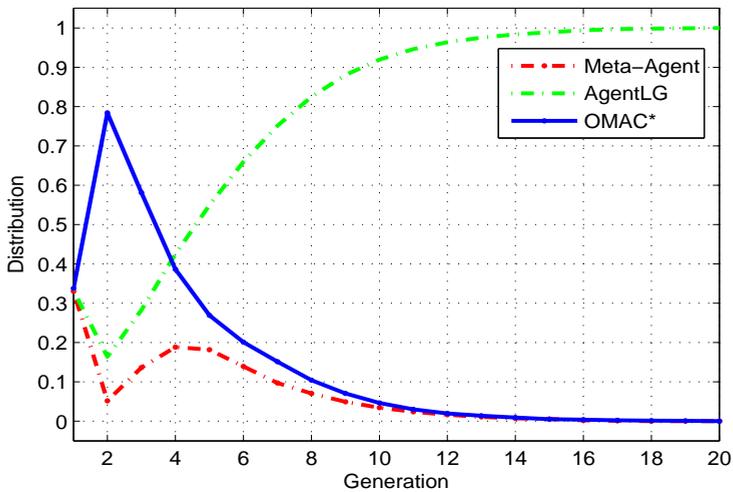


Figure 6.5: Strategy distributions over generations when players are allowed to interact with their neighbors.

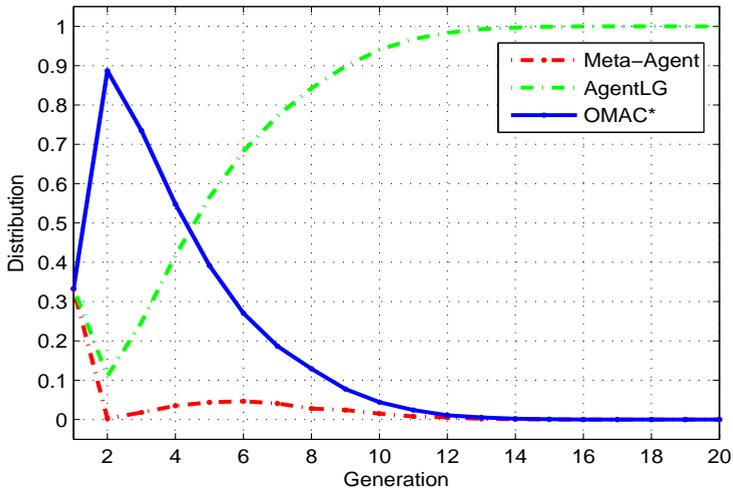


Figure 6.6: Strategy distributions over generations when players are allowed to interact with their neighbors.

6.3 Dynamics of All Strategies with Local Interaction

In this section, we turn our attention to the dynamics of a wider range of strategies. In more detail, we enlarge the strategy space to nine strategies considered in Table 5.1 and investigate the fitness of these strategies in a highly dynamic environment, in which numerous players have freedom to change its current strategy and mimic the most successful one in their locality. The result with respect to location between nearest neighbors is given in Figure 6.7. As can be seen, other strategies except OMAC* and Meta-Agent in a short period became extinct. Meta-Agent had a jump start, but its territory was invaded by OMAC* after 3 generations; the evolution finally developed into a state of local oscillation between them, where OMAC* achieved around 43% of the share in the end.

Then, we consider the case where the interaction is extended to next-nearest neighbors. The results are demonstrated in Figure 6.8. It is easy to find out that the game terminates in a state where OMAC* and Meta-Agent co-exist, with more apparent oscillation being observed. As a matter of fact, the frequencies of both two strategies used by players moved around 50%, in other words, OMAC* invaded a small proportion of Meta-Agent in one step, while Meta-Agent occupied that territory next step and this process continued. When extending interaction range of an agent to all other participants in the game, the results were given in Figure 6.9. The evolution at

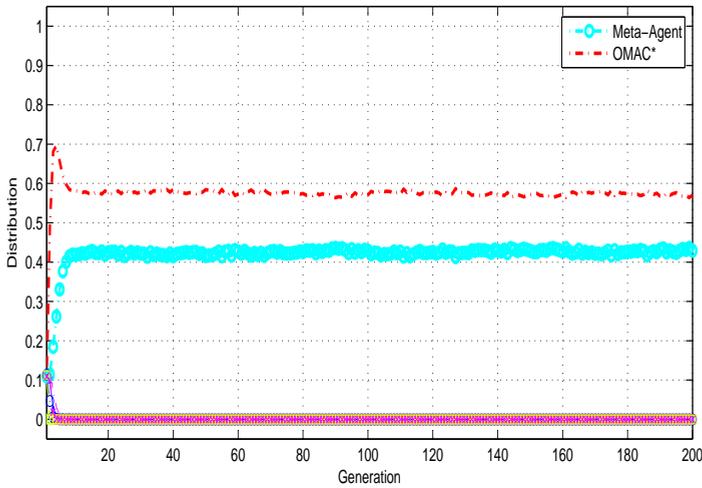


Figure 6.7: Strategy distributions over generations when players are allowed to interact with their neighbors.

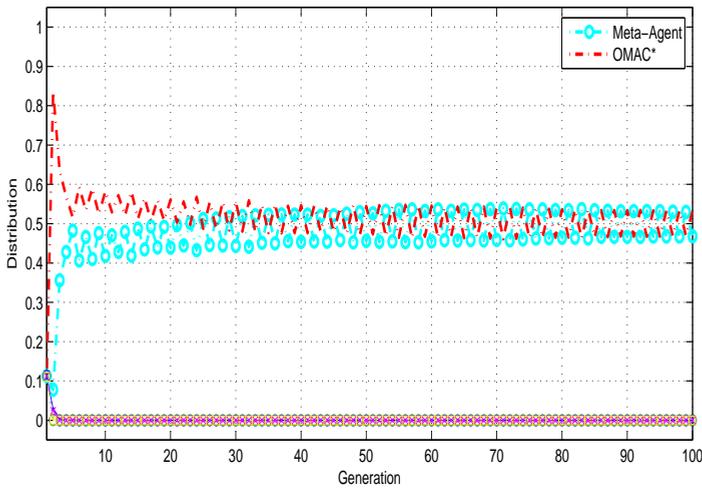


Figure 6.8: Strategy distributions over generations when players are allowed to interact with their neighbors' neighbors.

this case was terminated in a homogeneous state of OMAC* after a transient process, during which other competitors died out.

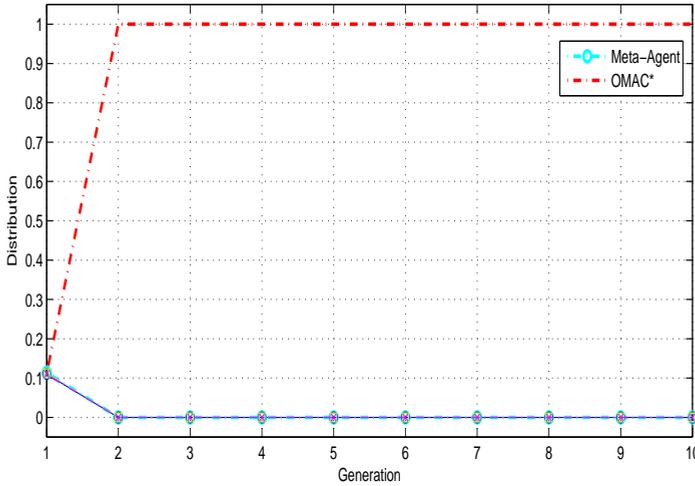


Figure 6.9: Strategy distributions over generations when players are allowed to interact with their neighbors' neighbors.

6.4 Discussion

In this chapter, we conducted performance analysis of the negotiation strategies on connected social graphs which specify both the game theoretical interaction between players and the learning mechanism, with the equal weight of each edge. Moreover, we investigate the effect of varying interaction ranges on evolution dynamics of population choices over these strategies. For the sake of comparison with empirical game-theoretical analytical results, we classify analysis into two parts – evolution dynamics over small and large strategy space, respectively.

Regarding the first three-strategy set, Dragon was replaced by the other two candidates in this three-strategy spatial game at a very early stage, regardless of interaction range. The absence of Dragon can also be observed in the EGT analysis as previously shown in Fig 5.2. This seemingly suggest Dragon is not suitable for a dynamic and open environment. This spatial system evolved into a pattern of oscillations; while in EGT analysis, no such oscillations did not happen between OMAC* and AgentLG (like a best cycle in EGT). OMAC*, given a short interaction range, could gradually increase its share, and eventually obtained like 57% of the population with small (local) oscillations. When considerations were taken into account a wider interaction range allowing agents negotiate with those agents in the area covering up to its next-nearest neighbors, OMAC* delivered a less satisfying performance, and Meta-Agent

turned out to be the most successful strategy. Moreover, both strategies' shares oscillated with a greater amplitude. This was due to a weaker self-play performance of OMAC* related to Meta-Agent, thus hampering its invasions of more territory. For the other small strategy set, the results were quite identical with those suggested by EGT analysis.

When extending the strategy pool to the whole range of strategies as given in Section 6.3, some phenomena similar to the first case were observed. In more detail, as evolution continued, OMAC* and Meta-Agent together managed to occupy the whole territory. Due to similar reasons as explained above, a wider interaction range however caused more obvious oscillations and affected OMAC*'s performance.

Part IV

Concluding Part

7

Contributions and Future Work

In Chapters 1 and 2 this thesis introduced key challenges associated with automated negotiation in complex environments and provided an overview of relevant related work. Then, in Chapters 3 to 6, it addressed the four identified research questions from different perspectives. This chapter first summarizes the contributions made with respect to the research questions and then discusses possible directions for future research.

7.1 Contributions to the Research Questions

In Chapters 3 and 4 several novel negotiating strategies were proposed that address Research Questions 1 and 2:

Research Question 1: *How to learn opponent models in complex negotiations?*

Current approaches tend not to be effective in complex environments. In order to handle the uncertainty of complex negotiations, it is essential to utilize negotiation schemes that can model unknown opponents under strict time constraints. Toward this end, several new learning-based approaches were developed that enable an agent to learn opponent strategies with respect to the prediction of the opponents' future concessions. Specifically, these approaches, which exploit decomposition techniques (as done by OMAC and EMAR) and Gaussian processes (as done by OMAC* and Dragon), are appropriate for complex negotiations because they are neither relying on simplifying assumptions about the negotiation uncertainty nor requiring a practically intractable computational load. They share an interesting and important feature, namely, to provide a computationally affordable opponent model despite the lack of prior knowledge about opponents and about the structure of contracts under negotiation.

These approaches are not only appropriate with respect to learning of opponent models, but also provide an answer to the second research question:

Research Question 2: *How to concede appropriately in environments of high uncertainty and dynamics?*

This question was addressed twofold. First, by aiming at effective concession strategies (to determine the amount of concession). In complex negotiations, an agent's opponents typically exhibit sophisticated behavior with the goal to avoid own concessions. As a negative effect, (direct) predictions of an agent can easily result in wrong expectations about the opponents' future willingness to make concessions. This, in turn, raises the risk for an agent to make higher concessions than necessary. Two concepts were introduced – the dynamic conservative expectation function ($R(t)$) and the expected received utility function ($E(t)$) – that keep this risk of “friendly concession making” as low as possible (given an agent's currently available opponent model). Whereas the negotiation strategies proposed in Chapters 3 and 4 employ these two concepts in different ways (e.g., whether or not to take into account the lowest expectation of the eventual benefit (as considered by EMAR) or the compromise point (as considered by Dragon)), they have in common that the decision on concession making is done in a highly flexible, dynamic manner.

Second, by aiming at effective offer-generating mechanisms. This is essential because in the case of multiple issues a concession made by an agent does not necessarily result in a higher satisfaction of the negotiating opponent. Several mechanisms, having different computational requirements, for selecting counter-offers were proposed: this includes mechanisms that generate as many offers as possible within a certain concession range in order to explore the possibly very large outcome space in limited time (see the offer-generating mechanisms of OMAC and EMAR described in 3.1.3 and 3.2.2); and it includes an ϵ -greedy strategy that supports a negotiating agent in appropriately choosing between opponent-exploiting and random offers (see the offer-generating mechanisms of OMAC* and Dragon presented in 4.1.4 and 4.2.4).

In Chapter 5 the third Research Question was addressed:

Research Question 3: *In addition to measuring the performance of negotiation strategies in tournament-based (thus fixed) negotiation scenarios, how to analyze the performance of negotiating agents in dynamic scenarios?*

Based on empirical game theory (EGT), three different types of dynamic scenarios of increasing complexity were designed to conduct investigations into strategy robustness. In more detail, EGT was employed to identify pure Nash equilibria or best reply cycles in the following cases:

Case 1: Single negotiation encounters between two players.

Case 2: Scenarios composed of six players with the top three strategies.

Case 3: Scenarios with a full combination of players and strategies.

These cases, which cover a wide range of practical negotiation scenarios, were used to carry out a comprehensive strategy analysis in repeated negotiation games. The results show how attractive a strategy can be in the context of other strategies for a group of agents that can choose dynamically among the given strategies. From a more general perspective, the results show that the proposed EGT-based performance analysis is well suited for dynamic negotiation settings.

A shortcoming of EGT-based analysis is that they are not suited for evaluating the performance of strategies in settings where interaction locality has to be taken into consideration and the number of participants can be large. This led to the fourth Research Question, which was addressed in Chapter 6:

Research Question 4: *How to analyze the performance of a negotiation strategy in scenarios where a large population of negotiating autonomous agents is involved and the interaction range of the agents may be restricted?*

This question was answered by employing techniques of evolutionary game theory. To construct a spatial negotiation model, we defined the interaction network (connectivity structure) through a hexagon lattice where the edges connect direct “negotiating neighbors”. Moreover, a general heuristic strategy-updating rule for players was employed that allows each player to mimic the most successful strategy in its interaction range. Based on this generic model we performed a detailed performance analysis on various games with different sizes of strategy space and different ranges of agent interaction. The achieved results make it possible to characterize the overall behavioral profile of the population such that the evolution of strategy shares in a broad range of situations can be determined. In essence, these shares correspond to indicators of the overall fitness of locally played strategies in large-scale negotiation settings.

Overall, the research conducted in response to the addressed research questions provides a comprehensive answer to the underlying problem statement and the two subproblems induced by it.

7.2 Directions for Future Work

As described above, the research presented in this thesis has advanced the state-of-the-art of complex negotiations in several aspects. Moreover, it also opens various

new, interesting and challenging research directions. In the following, we discuss some directions which we consider as most promising and relevant.

Interdependent multi-issue domains: The focus of the proposed negotiation approaches is on negotiations in which the negotiated issues are independent of each other. Linear utility functions are therefore used to describe the agents' preferences over the different issues. However, for many real-world negotiations the issues can be interdependent rather than independent: this holds whenever an agent's choice of one issue is affected by choices it made about other issues. Such interdependencies result in non-linear utility functions and do further complicate the decision-making process. To extend the proposed approaches toward interdependent multi-issue negotiation scenarios is an important topic for further research.

Knowledge transfer between tasks: Learning an opposing agent's behavior model is hard because its behavior can only be observed indirectly through offers refused and counter offers made by the opposing agent. This obviously hampers the efficacy with which opponent models can be learnt. Transfer learning, as an important emerging branch of machine learning, enables the use of knowledge learnt in a so called source task to aid learning in a different yet related target task. Transfer is substantially more demanding than simply using the previous history encountered by a negotiating agent. Specifically, if the source task and the target task belong to different negotiation domains, then transfer learning requires an appropriate, very carefully designed (or learned) mapping of "source knowledge" to "target knowledge". The benefit of this mapping would be that an agent has additional information at hand so that it can learn more efficiently to negotiate successfully in a new domain. In other words, knowledge transfer allows to accelerate the learning process and to reach the same or better performance with less learning instances. The exploration of this kind of knowledge transfer in the context of automated negotiation constitutes another promising avenue of research.

Concurrent negotiation: Rather than interacting with only one opponent at a time (i.e., sequential negotiation), concurrent negotiation requires from agents to negotiate with several or even all involved negotiating opponents at the same time. Agents, in this case, can receive a number of offers from different types of opponents in a short period of time, which creates novel possibilities with respect to learning of opponent models and negotiating in a highly flexible way. On the other hand, concurrent negotiation is still poorly understood and it is unclear how to take advantage of these possibilities. An interesting question thus is whether and to what extent the negotiation techniques explored in this thesis can be modified and extended so that

they are of value for concurrently negotiating agents.

Agents against humans negotiators: This thesis dealt only with agent-agent negotiation. Another important, practically relevant type of negotiation are “hybrid” negotiations in which both agents and humans are involved. Evidently, human experts are more flexible and sophisticated compared to automated negotiators. In such a hybrid setting, learning of opponent models is extremely challenging because a human opponent may change his or her negotiation strategy also during an ongoing session. Moreover, the number of rounds tend to be limited for an agent-human negotiation setting (because of the longer response time needed by humans), which puts additional pressure on an agent to propose offers that are likely to be accepted by a human negotiator. Last but not least, hybrid settings are challenging because humans do not necessarily negotiate fully rationally, whereas the strategies and methods developed for automated negotiations are implicitly based on the assumption that the opponents always act rationally. Playing against human negotiators requires to give up the idea of full rationality and poses particularly high demands on the learning abilities of an automated negotiator. As OMAC* is particularly strong in these abilities, it appears to be a promising starting point for human-machine negotiations. It would be interesting to find out how well OMAC* (equipped with an appropriate communication interface) performs when playing against human negotiators. Research along this line can lead to valuable insights w.r.t. both the design of automated negotiation strategies and the strategic behavior of human negotiators.

Bibliography

- [1] The Third International Automated Negotiating Agent Competition (ANAC 2012). <http://anac2012.ecs.soton.ac.uk/results/final/>, 2012.
- [2] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. New York, USA, 1965.
- [3] Sunil Adhau, M.L. Mittal, and Abhinav Mittal. A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, 25(8):1738 – 1751, 2012.
- [4] Bo An, Victor Lesser, David Irwin, and Michael Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 981–988. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
- [5] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen V. Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73 – 103, 2013.
- [6] Tim Baarslag and Koen V Hindriks. Accepting optimally in automated negotiation with incomplete information. In *Proceedings of AAMAS'2013*, pages 715–722. ACM, 2013.
- [7] Tim Baarslag, Koen V. Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. *The First Automated Negotiating Agents Competition (ANAC 2010)*, volume 383 of *Studies in Computational Intelligence*, pages 113–135. Springer-Verlag, 2012.
- [8] Carrie Beam and Arie Segev. Automated negotiations: A survey of the state of the art. *Wirtschaftsinformatik*, 39(3):263–268, 1997.
- [9] George Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control, third edition*. Prentice-Hall, 1994.
- [10] Jakub Brzostowski and Ryszard Kowalczyk. Predicting partner’s behaviour in agent negotiation. In *Proceedings of the Fifth Int. Joint Conf. on Autonomous*

- Agents and Multiagent Systems*, pages 355–361, New York, NY, USA, 2006. ACM.
- [11] Réal Carbonneau, Gregory E. Kersten, and Rustam Vahidov. Predicting opponent’s moves in electronic negotiations using neural networks. *Expert Syst. Appl.*, 34:1266–1273, February 2008.
- [12] Siqi Chen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. Optimizing complex automated negotiation using sparse pseudo-input Gaussian processes. In *Proceedings of the 12th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 707–714. IFAAMAS, 2013.
- [13] Siqi Chen and Gerhard Weiss. An efficient and adaptive approach to negotiation in complex environments. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 228–233. IOS Press, 2012.
- [14] Siqi Chen and Gerhard Weiss. A novel strategy for efficient negotiation in complex environments. In *Proceedings of the 10th German Conference on Multiagent System Technologies*, pages 68–82. Springer-Verlag, 2012.
- [15] Siqi Chen and Gerhard Weiss. An efficient automated negotiation strategy for complex environments. *Engineering Applications of Artificial Intelligence*, 26(10):2613 – 2623, 2013.
- [16] Siqi Chen and Gerhard Weiss. An approach to complex agent-based negotiations via effectively modeling unknown opponents (Under review). 2014.
- [17] Siqi Chen and Gerhard Weiss. An intelligent agent for bilateral negotiation with unknown opponents in continuous-time domains (In press). *ACM Transactions on Autonomous and Adaptive Systems*, 9(3), 2014.
- [18] Robert M. Coehoorn and Nicholas R. Jennings. Learning on opponent’s preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th Int. conf. on Electronic commerce*, pages 59–68, New York, NY, USA, 2004. ACM.
- [19] Jiangbo Dang and Michael N. Huhns. Concurrent multiple-issue negotiation for internet-based services. *IEEE Internet Computing*, 10:42–49, 2006.
- [20] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 2006.
- [21] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

- [22] Lei Duan, Mustafa K. Dogru, Ulas Ozen, and J.Christopher Beck. A negotiation framework for linked combinatorial optimization problems. *Autonomous Agents and Multi-Agent Systems*, 25(1):158–182, 2012.
- [23] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Negotiation decision functions for autonomous agents. *Rob. Autom. Syst.*, 24(4):159–182, 1998.
- [24] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.
- [25] Patrick Flandrin, Gabriel Rilling, Paulo Goncalves, and I. Emd Basics. Empirical mode decomposition as a filter bank. *IEEE Signal Proc Lett.* 11 : 112-114., 2004.
- [26] Katsuhide Fujita, Takayuki Ito, Tim Baarslag, Koen V. Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. *The Second Automated Negotiating Agents Competition (ANAC2011)*, volume 435 of *Studies in Computational Intelligence*, pages 183–197. Springer Berlin / Heidelberg, 2013.
- [27] Carla P Gomes. Computational sustainability: Computational methods for a sustainable environment, economy, and society. *The Bridge*, 39(4):5–13, 2009.
- [28] Jianye Hao and Hofung Leung. ABiNeS: An adaptive bilateral negotiating strategy over multiple items. In *Proceedings of WI/IAT’2012*, pages 95–102. IEEE Computer Society, 2012.
- [29] Mark Hendrikx. A survey of opponent models in automated negotiation. Technical report, Delft University of Technology, The Netherlands, September 2011.
- [30] K. V. Hindriks, C. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proceedings of AAMAS’2009*, pages 1397–1398. ACM, 2009.
- [31] Koen V. Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov. The benefits of opponent models in negotiation. In *Proceedings of WI/IAT’2009*, volume 2, pages 439–444. IET, 2009.
- [32] Koen V. Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 331–338, Estoril, Portugal, 2008. ACM.
- [33] Chongming Hou. Predicting agents tactics in automated negotiation. In *Intelligent Agent Technology, IEEE / WIC / ACM International Conference on*, volume 0, pages 127–133, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

- [34] Norden E. Huang and Samuel S.P. Shen. *Hilbert-Huang transform and its applications*. World Scientific, 2005.
- [35] Norden E. Huang, Man-Li Wu, Wendong Qu, Steven R. Long, and Samuel S.P. Shen. Applications of hilbert-huang transform to non-stationary financial time series analysis. *Appl. Stoch. Models Bus. Ind.*, 19(3):245–268, 2003.
- [36] Norden E. Hunag, Zheng Shen, and Steven R. Long. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. A*, pages 903–995, 1998.
- [37] Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors. *New Trends in Agent-based Complex Automated Negotiations*. Springer-Verlag, 2012.
- [38] Takayuki Ito, Minjie Zhang, Valentin Robu, and Tokuro Matsuo, editors. *Complex Automated Negotiations: Theories, Models, and Software Competitions*. Springer Berlin / Heidelberg, 2013.
- [39] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.
- [40] CatholijnM. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
- [41] Patrick R. Jordan, Christopher Kiekintveld, and Michael P. Wellman. Empirical game-theoretic analysis of the TAC supply chain game. In *Proceedings of AAMAS’2007*, pages 1188–1195. ACM, 2007.
- [42] Ehud Kalai and Meir Smorodinsky. Other solutions to nash’s bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 513–518, 1975.
- [43] Timothy Killingback and Michael Doebeli. Spatial evolutionary game theory: Hawks and doves revisited. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1374):1135–1144, 1996.
- [44] Guoming Lai, Cuihong Li, Katia Sycara, and Joseph Andrew Giampapa. Literature review on multi-attribute negotiations. Technical report, Robotics Institute, Carnegie Mellon University, December 2004.
- [45] 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, 2008.

- [46] Raz Lin and Sarit Kraus. Can automated agents proficiently negotiate with humans? *Communications of the ACM*, 53(1):78–88, 2010.
- [47] 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*, 172:823–851, April 2008.
- [48] Lennatt Ljung. *System Identification: Theory for the User*. NUpper Saddle River, NJ, Prentice-Hal PTR, 1999.
- [49] Alessio R Lomuscio, Michael Wooldridge, and Nicholas R Jennings. A classification scheme for negotiation in electronic commerce. *Group Decision and Negotiation*, 12(1):31–56, 2003.
- [50] Fernando Lopes, Michael Wooldridge, and A. Novais. Negotiation among autonomous computational agents: principles, analysis and challenges. *Artificial Intelligence Review*, 29:1–44, 2008.
- [51] Michael Luck and Peter McBurney. Computing as interaction: Agent and agreement technologies. In *Proc. IEEE SMC Conference on Distributed Human-Machine Systems*, pages 1–6, 2008.
- [52] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Takayuki Ito, Minjie Zhang, Quan Bai, and Katsuhide Fujita, editors. *Novel Insights in Agent-based Complex Automated Negotiation* (In Press). Springer, 2013.
- [53] Peter Mell and Tim Grance. The nist definition of cloud computing. 2011.
- [54] Yishay Mor, Claudia V. Goldman, and Jeffrey S. Rosenschein. Learn your opponent’s strategy (in polynomial time)! In *In Proceedings of IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems*, pages 164–176. Springer-Verlag, 1996.
- [55] Abhinay Muthoo. *Bargaining theory with applications*. Cambridge University Press, 1999.
- [56] Roger B Myerson. Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica: Journal of the Econometric Society*, pages 883–897, 1981.
- [57] John Nash. Two-person cooperative games. *Econometrica*, 21(1):pp. 128–140, 1953.
- [58] John F Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

- [59] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [60] Yinon Oshrat, Raz Lin, and Sarit Kraus. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 377–384. International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [61] Sanghyun Park and Sung-Bong Yang. An efficient multilateral negotiation system for pervasive computing environments. *Eng. Appl. Artif. Intell.*, 21(4):633 – 643, 2008.
- [62] Ilja Ponka. *Commitment models and concurrent bilateral negotiation strategies in dynamic service markets*. PhD thesis, University of Southampton, School of Electronics and Computer Science, 2009.
- [63] Azzurra Ragone, Tommaso Noia, Eugenio Sciascio, and Francesco M. Donini. Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces. *Autonomous Agents and Multi-Agent Systems*, 16(3):249–270, 2008.
- [64] Iyad Rahwan, Ryszard Kowalczyk, and Ha Hai Pham. Intelligent agents for automated one-to-many e-commerce negotiation. In *Australian Computer Science Communications*, volume 24, pages 197–204. Australian Computer Society, Inc., 2002.
- [65] Iyad Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(04):343–375, 2003.
- [66] Howard Raiffa. *The art and science of negotiation*. Harvard University Press Cambridge, Mass, 1982.
- [67] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [68] Prashant P Reddy and Manuela M Veloso. Negotiated learning for smart grid agents: Entity selection based on dynamic partially observable features. In *AAAI*, 2013.
- [69] D.K. Ruch and P.J.V. Fleet. *Wavelet theory: an elementary approach with applications*. John Wiley & Sons, 2009.

- [70] Sabyasachi Saha, Anish Biswas, and Sandip Sen. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the Fourth international joint conference on Autonomous agents and multiagent systems*, pages 397–403, New York, NY, USA, 2005. ACM.
- [71] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2009.
- [72] Carles Sierra, Vicente J. Botti, and Sascha Ossowski. Agreement computing. *Kunstliche Intelligenz*, 25(1):57–61, 2011.
- [73] Carles Sierra, Nick R Jennings, Pablo Noriega, and Simon Parsons. A framework for argumentation-based negotiation. In *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pages 177–192. Springer, 1998.
- [74] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT press, 2006.
- [75] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
- [76] György Szabó and Gábor Fáth. Evolutionary games on graphs. *Physics Reports*, 446(4):97–216, 2007.
- [77] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton university press, 2007.
- [78] Minhong Wang, Huaiqing Wang, Doug Vogel, Kuldeep Kumar, and Dickson K.W. Chiu. Agent-based negotiation and decision making for dynamic supply chain formation. *Eng. Appl. Artif. Intell.*, 22(7):1046 – 1055, 2009.
- [79] Gerhard Weiss, editor. *Multiagent Systems, 2nd edition*. MIT Press, 2013.
- [80] Michael P. Wellman. Methods for empirical game-theoretic analysis. In *Proceedings of the 21st national conference on Artificial intelligence*, pages 1552–1555, Boston, Massachusetts, 2006. AAAI Press.
- [81] Colin R Williams. *Practical Strategies for Agent-Based Negotiation in Complex Environments*. PhD thesis, University of Southampton, 2012.
- [82] C.R. Williams, V. Robu, E.H. Gerding, and N.R. Jennings. Using gaussian processes to optimise concession in complex negotiations against unknown opponents. In *Proceedings of IJCAI’2011*, pages 432–438. AAAI Press, 2011.

- [83] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [84] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.
- [85] HP Young. The evolution of conventions. *Econometrica: Journal of the Econometric Society*, pages 57–84, 1993.
- [86] Shahriar Yousefi, Ilona Weinreich, and Dominik Reinartz. Wavelet-based prediction of oil prices. *Chaos Solitons Fractals*, 25(2):265–275, 2005.
- [87] Lean Yu, Shouyang Wang, and Kin Keung Lai. Forecasting crude oil price with an emd-based neural network ensemble learning paradigm. *Energy Economics*, 30(5):2623 – 2635, 2008.
- [88] Dajun Zeng and Katia Sycara. Benefits of learning in negotiation. In *Proceedings of AAAI'1997*, pages 36–41. AAAI Press, 1997.
- [89] Dajun Zeng and Katia Sycara. How can an agent learn to negotiate? In *Intelligent Agents III Agent Theories, Architectures, and Languages*, pages 233–244. Springer, 1997.
- [90] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48(1):125–141, January 1998.

List of Figures

3.1	4-level decomposition using DB10	28
3.2	The approximation part of an example concession curve.	30
3.3	Predictive power of OMAC	31
3.4	Raw Scores of OMAC and benchmark agents in the domains.	36
3.5	A brief flowchart of EMAR.	38
3.6	Decomposition example of the EMD approach	41
3.7	Prediction power of EMAR	42
3.8	Raw scores of EMAR and benchmark agents in all domains.	48
3.9	Performance of EMAR and other agents w.r.t three reservation values.	51
4.1	Performance of OMAC* and others w.r.t various opposition levels.	67
4.2	Performance of OMAC* and others in the largest domains.	68
4.3	Performance comparison between OMAC* and others.	72
4.4	Comparison between GP and SPGPs	75
4.5	Normalized scores of Dragon and others in the primary tournament.	83
4.6	Normalized scores of Dragon and others in the advanced tournament.	84
4.7	Performance of Dragon and others w.r.t various competitiveness levels.	85
4.8	Performance of Dragon and others w.r.t various domain sizes.	86
5.1	Deviation analysis of two-player encounters.	94
5.2	Deviation analysis of a three-player with three-strategy scenario.	96
5.3	Deviation analysis of a three-player with another three-strategy scenario.	97
5.4	The deviation analysis of a nine-player with nine-strategy scenario	98
6.1	A locality example in a hexagon model.	103
6.2	An example of strategy distribution evolution.	104
6.3	Distributions of three strategies I	105
6.4	Distributions of three strategies II.	106
6.5	Distributions of another three strategies I.	106
6.6	Distributions of another three strategies II.	107
6.7	Distributions of nine strateiges I.	108
6.8	Distributions of nine strateiges II.	108
6.9	Distributions of nine strateiges III.	109
7.1	Negotiation-based e-commerce.	132

7.2	Clouding computing with automated negotiation.	134
7.3	Smart grid with automated negotiation.	135

List of Tables

3.1	Normalized scores of OMAC and benchmarking agents.	37
3.2	Overall performance of OMAC and benchmarking agents.	37
3.3	Test domains for evaluating EMAR.	46
3.4	Average raw score of EMAR and benchmarking agents.	47
3.5	Overall performance of EMAR and benchmarking agents.	49
3.6	Overall performance of EMAR and others averaged across all domains.	52
4.1	Overview of application domains for OMAC*.	63
4.2	Overview of benchmark agents for evaluating OMAC*.	64
4.3	Overview of primary parameter settings for OMAC*.	65
4.4	RMS errors of various learning approaches.	66
4.5	Performance of OMAC* and others w.r.t various opposition levels.	67
4.6	Performance of OMAC* and others w.r.t various discounting factors.	69
4.7	Performance of OMAC* and others w.r.t various reservation values.	70
4.8	Overall performance of OMAC* and the competitors.	71
4.9	Overview of test domains for evaluating Dragon.	82
4.10	Overall performance of Dragon and its competitors.	87
5.1	Payoff matrix involving of top strategies.	93

Addendum: Valorization

Remark: This Addendum is required by the Regulation governing the attainment of doctoral degrees (2013) of Maastricht University. As stated there, the addendum “does not form part of the dissertation and should not be assessed as part of the dissertation”.

Because of the rapid progress that has been made in the field of automated negotiation in the previous five to ten years, various interesting valorization and application options are currently emerging in diverse domains and areas. Below four such domains are described of which we think they possess a particularly high valorization potential for automated negotiation techniques and methods.

Electronic Commerce: Nowadays electronic commerce, or e-commerce for short, is a standard trading form in many areas of economy. E-commerce means that businesses and individuals buy and sell products and services via electronic media and especially through the internet and other computer networks. Three types of e-commerce usually distinguished are business to business (B2B), business to consumer (B2C), and consumer to consumer (C2C). For all these types the focus has mainly been on building an appropriate infrastructure for facilitating business transactions and the exchange of electronic information. As a result of the tremendous advantages of e-commerce such as lower costs, convenience and higher speed of trading (compared to conventional paper-based trading), various new types of innovative businesses have appeared and sustained, such as Amazon (a worldwide B2C online retailer), Jingdong (one of the leading B2C online retailers in China) and Alibaba (the largest B2B marketplace in China). However, the operational effectiveness that would be needed to meet the high demands on e-commerce has not yet been achieved. Instead, current technologies used in e-commerce merely target at reliably facilitating relatively simple business transactions, whereas they fall short when it is about to support an effective and efficient automation of non-trivial business decisions [64].

This shortcoming of e-commerce apparently concerns (among other things) human-human negotiation processes, because these processes can be very time-consuming and thus costly. In fact, automated negotiation without human intervention, including the automation of underlying decision-making processes, is not yet solved in the context of electronic trading. This results in a growing pressure to explore powerful solutions for such an automation. Agent-based negotiation, as

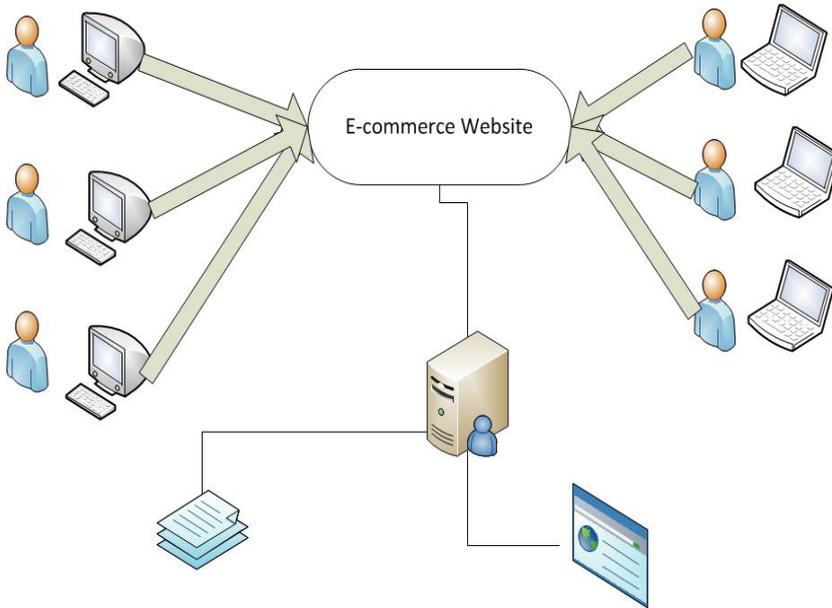


Figure 7.1: Sample setting of negotiation-based e-commerce: multiple human traders coordinate their buying-selling activities electronically through a shared website.

considered and proposed in this thesis, is an obvious solution candidate and a promising method for supporting high-level automation of business decisions. Thereby it is particularly suited as a tool for automating trading situations in which two or more parties multilaterally bargain resources for the purpose of achieving mutual gain [8]. Figure 7.1 illustrates the e-commerce domain.

Multi-agent Systems: There are a several interesting valorization opportunities in the context of commercial applications of multi-agent systems. Multi-agent systems are, in a nutshell, computerized systems composed of multiple intelligent agents (e.g., computers, softwares or programmes) that act and interact intelligently and autonomously in a possible very complex environment. Agent-based negotiation, as a powerful fundamental communication mechanism, can be of major help regarding cooperation and coordination in multi-agent settings. One of core problems in such settings is that resource allocation (i.e., the assignment of resources to the individual agents) often can not be done in a centralized way due to various practical constraints such as computational overhead, limited system-wide communication, privacy issues, and real-time requirements. For example, one may think of the task of allocating the sensors (as the resources) being part of a wireless sensor network to agents that are

responsible for certain surveillance activities [4]. Rather than trying to implement resource allocation centrally, allocation can be done in a distributed style by means of local automated negotiations among autonomous agents. Potential advantages would be increased robustness and scalability and (if concurrent negotiations are allowed) also a speed-up in agreement finding and conflict resolution. Another example of a multiagent-system application for which the concepts and approaches proposed in this thesis are relevant is supply chain management (especially when organized in form of an electronic marketplace or a network of market places). Here automated negotiation can be very helpful to constitute an effective mechanism for coordinating decisions between firms by suggesting mutually beneficial contracts.

Cloud Computing: Cloud computing provides a specific network-based computing model where a programme or application can run on a connected server (or on several servers) rather than individually on a local computing device such as a computer, a tablet or a smartphone. In a cloud environment, customers rent one or several computing resources from remote servers in such a way that overall coherence and economic viability is achieved [53]. (These intended effects are similar to those in producer-consumer networks such as the electricity grid; see also below.) However, cloud resources are not simply shared by a fixed set of users, but tend to be dynamically reallocated per demand in a real-time fashion. This is because the design objective of cloud computing, in general, aims at maximizing the effectiveness of the overall shared resources instead of a part of consumers' utility. Using automated negotiation, consumers can effectively reach agreements among resource providers for networked resource lease in a dynamic and uncertain market (e.g., the environment of cloud computing). A consumer thus would be enabled to establish new contracts and de-commit from existing ones in a very flexible way, which means it would be free for he to make a decommitment decision (with penalty) if he finds another contract with better conditions. Furthermore, negotiation-based models also have the potential to achieve a higher social welfare compared with auction-based models in scenarios as they could be found in cloud computing environments [4]. Figure 7.2 illustrates the cloud-computing domain.

Smart Grid: An integration of distributed renewable energy resources, e.g., wind, water and solar, into electricity supply system is essential to alleviate the impact of the growing energy demand into environment [27]. Such energy distribution concepts, known as smart (energy) grids, are of very special value for developing countries that are rich in sustainable natural energy resources while lacking fossil fuels (e.g., coal, petroleum and gas). However, the introduction of these resources to power supply system often results in significant volatility to the level of power supply, and thus

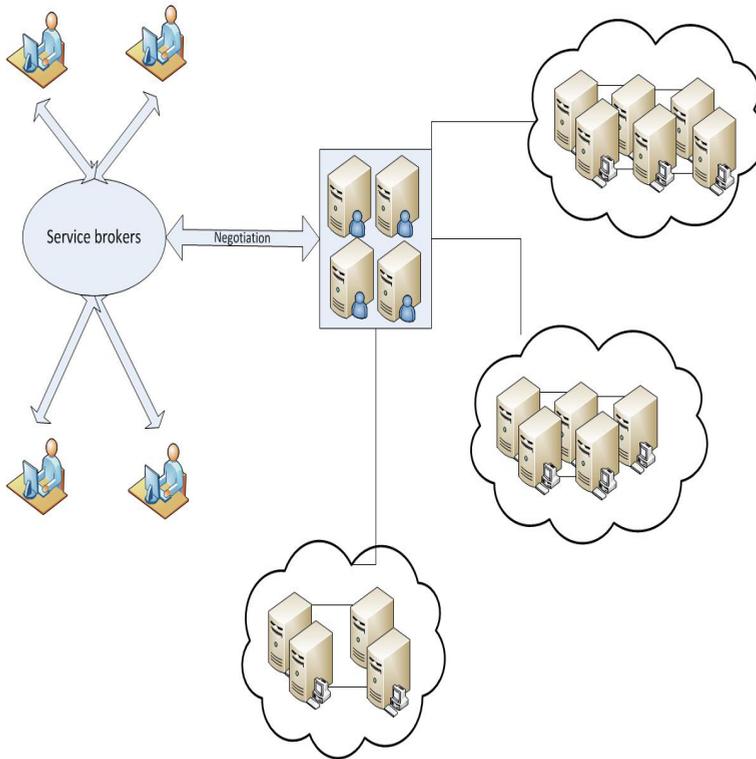


Figure 7.2: Negotiation-based flexible resource allocation in cloud computing. The figure illustrates a scenario where a host of clients negotiate with service brokers over leasing remote computing resources that are geographically distributed (e.g., they may be located in different cities or countries).

makes it difficult to match power supply with client demand [68]. In response to this critical problem, agent-based negotiation can be applied in smart grid systems so that power generation and consumption is better balanced in a dynamic, adaptive and efficient manner. For example, an autonomous negotiating agent can assist a human customer in adaptively negotiating rapidly changing tariffs with various suppliers, while at the same time supporting the customer to adjust her demand (i.e., to change her preferences) in response to changes in the tariffs offered on the power market. Figure 7.3 illustrates the smart-grid domain.

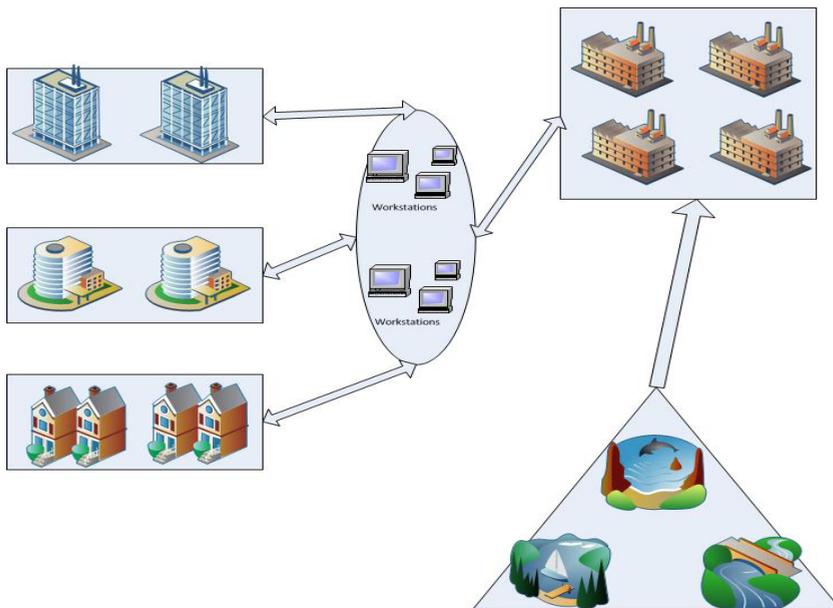


Figure 7.3: Illustration of a smart grid enhanced with automated negotiation, in which multiple types of power consumers (e.g., buildings, hotels and houses) and power sources (e.g., solar, wind, rivers and oceans) are networked. Power factories generate electricity from these sources and then deliver it to the market with varying tariffs. Negotiator agents bargain on behalf of human customers over tariff contracts with suppliers to reach favorable agreements, thereby taking into account customer preferences.

Summary

Over the past decade automated negotiation among autonomous software agents (agents for short) has developed into a topic of central interest in the area of intelligent agents and multi-agent systems. The main reason for this development is the broad range of applicability of automated negotiations in fields such as electronic commerce, electronic markets, supply chain management and service allocation. This thesis focuses on complex automated negotiation settings that show the following three challenging characteristics. First, the negotiating agents do not have knowledge about each other (i.e., they have not encountered before) and, in particular, do not have any prior information about the negotiation preferences or strategies of their respective opponents. Second, negotiation is being executed with deadline and with discount. This means that the negotiation process runs under real-time constraints and the final utility decreases over time according to some given discounting factor. And third, computational efficiency is important because agents (e.g., an agent running on a mobile device) may have very limited computing resources. These characteristics, which are very common in practical negotiations, impose high demands on the negotiation abilities of agents.

The *problem statement* that guided the research described in this thesis is:

How to design effective strategies for such complex negotiation settings and how to evaluate these strategies effectively?

Four complementary research questions were derived from this problem statement:

- *Research Question 1:* How to learn opponent models in complex negotiations?
- *Research Question 2:* How to make appropriate concessions in environments of high uncertainty and dynamics?
- *Research Question 3:* In addition to measuring the performance of negotiation strategies in tournament-based (thus fixed) negotiation scenarios, how to analyze the performance of negotiating agents in dynamic scenarios?
- *Research Question 4:* How to analyze the performance of a negotiation strategy in scenarios where a large population of negotiating autonomous agents is involved and the interaction range of the agents may be restricted?

After introducing the theme of this thesis and providing relevant background information in Chapters 1 and 2, respectively, the Research Questions 1 and 2 are

addressed in Chapters 3 and 4. Four learning-based negotiation strategies – OMAC, EMAR, OMAC*, and Dragon – are presented that enable a negotiating agent to generate computationally affordable models of its opponents. These models serve as a basis for predicting the opponents' future negotiation behaviors and making effective decisions during the negotiation process. A main difference of these four strategies is in the modeling techniques they apply: OMAC uses discrete wavelet transformation, EMAR uses empirical mode decomposition, OMAC* combines discrete wavelet transformation and standard Gaussian processes, and Dragon relies on sparse pseudo-input Gaussian processes. With respect to concession making, these strategies provide different, effective solutions for two core requirements for successful negotiation, namely, the avoidance of premature concessions and the generation of counter-offers that are both attractive to the opponents and suited for exploring the negotiation space. Common to these strategies is that concessions are made in a highly flexible and dynamic style. Extensive experimental results are provided that demonstrate the qualities of these strategies. The results also show that they clearly outperform the best negotiation strategies from recent editions of the International Automated Negotiating Agents Competition (ANAC) in a broad range of negotiation scenarios.

The Research Questions 3 and 4 are addressed in Chapters 5 and 6, respectively. The currently most common evaluation criterion for negotiation strategies is the performance level achieved in simulated tournaments. This criterion, however, is not suited for the evaluation of strategies in dynamic negotiation scenarios, that is, in scenarios in which agents are allowed to change their strategies during negotiation. In Chapter 5 it is shown how empirical game theory (EGT) can be exploited to evaluate negotiation strategies in such scenarios. Moreover, tournament-based evaluation (as well as EGT-based evaluation) is not suited for evaluating negotiation strategies in scenarios in which a large number of agents are involved and in which the individual agents interact only locally rather than with all other agents. In Chapter 6 it is shown how spatial evolutionary game theory can be used to evaluate the performance of strategies in such scenarios.

Finally, Chapter 7 summarizes the contributions of this thesis and identifies promising directions for future research that are opened by the described research.

Overall, the research described in this thesis advances the state of the art in automated negotiation in two important ways. First, by developing several novel negotiation strategies that offer highly effective opponent-modeling and concession-making techniques and methods. Second, by showing how game-theoretic concepts and techniques can be used for strategy evaluation in practically relevant negotiation scenarios for which the traditional and currently most common tournament-based evaluation approach fails short.

Samenvatting

Het onderzoek in geautomatiseerde onderhandelingen tussen agenten heeft zich in het afgelopen decennium ontwikkeld tot een belangrijk onderwerp in het veld van intelligente agenten en multi-agent systemen. De hoofdreden hiervoor is de brede toepasbaarheid van geautomatiseerde onderhandelingen in verschillende domeinen, zoals e-commerce, elektronische markten, supply chain management en dienst-toewijzing. Dit proefschrift is gericht op complexe geautomatiseerde onderhandelingsomgevingen, welke de volgende drie uitdagende eigenschappen vertonen: Ten eerste, de onderhandelende agenten hebben geen voorafgaande kennis over elkaar, d.w.z. ze hebben mekaar niet vooraf ontmoet, en ze hebben geen toegang tot informatie over de onderhandelingsvoorkeuren of strategieën van de tegenstanders. Ten tweede, de onderhandelingen worden uitgevoerd met een deadline en met discontering. Dit houdt in dat het onderhandelingsproces wordt uitgevoerd met een reële tijdslimiet en dat het uiteindelijke uitkeringsbedrag mettertijd afneemt afhankelijk van een gegeven kortingsfactor. Ten derde, aangezien de agenten over een beperkte rekenkracht beschikken is computationele efficiëntie cruciaal (b.v. een software agent op een smartphone). Deze typische eigenschappen stellen hoge eisen aan de onderhandelingsvaardigheden van agenten.

Het onderzoek van dit proefschrift is gestuurd door de volgende probleemstelling:

Hoe kan men effectieve strategieën voor complexe onderhandelingsomgevingen ontwikkelen en hoe kunnen deze strategieën effectief geëvalueerd worden?

Uit de probleemstelling zijn vier aanvullende onderzoeksvragen afgeleid:

- *Onderzoeksvraag 1:* Hoe kan men tegenstandermodellen leren tijdens complexe onderhandelingen?
- *Onderzoeksvraag 2:* Hoe kan men passende concessies uitvoeren in dynamische omgevingen met grote onzekerheid?
- *Onderzoeksvraag 3:* Hoe kan men de prestaties van onderhandelingsstrategieën meten in zowel toernooi-gebaseerde onderhandelingsscenario's alsook in dynamische scenario's?
- *Onderzoeksvraag 4:* Hoe kan men de prestatie van een onderhandelingsstrategie meten in een scenario met een groot aantal autonome onderhandelende agenten welke maar over een beperkt interactiebereik beschikken?

Het onderwerp van dit proefschrift en relevante achtergrondinformatie worden in hoofdstukken 1 en 2 behandeld, waarop volgend de onderzoeksvragen 1 en 2 in hoofdstukken 3 en 4 behandeld worden. Vier leer-gebaseerde onderhandelingsstrategieën – OMAC, EMAR, OMAC*, en Dragon – worden gepresenteerd welke het voor een agent mogelijk maken om een computationeel efficiënt model van zijn tegenstanders te genereren. Deze modellen vormen de basis voor het voorspellen van het onderhandelgedrag van de tegenstanders en het sluiten van effectieve beslissingen tijdens het onderhandelingsproces. Het grote verschil tussen deze vier strategieën is de toegepaste modeleringstechniek: OMAC maakt gebruik van discrete wavelet transformaties, EMAR maakt gebruik van empirische mode-decompositie, OMAC* combineert wavelet transformatie met standard Gauss-processen, en Dragon past sparse pseudo-input Gauss-processen toe. Met betrekking tot het doen van concessies bieden deze strategieën effectieve oplossingen voor twee kernvereisten van een succesvolle onderhandeling: namelijk het vermijden van voortijdige concessies en het genereren van tegenaanbiedingen, dewelke zowel aantrekkelijk zijn voor de tegenstanders alsook geschikt zijn voor het verkennen van de onderhandelingsruimte. Deze strategieën hebben met elkaar gemeen dat concessies in een zeer flexibel en dynamische stijl worden uitgevoerd. Met experimentele resultaten worden de kwaliteiten van deze strategieën gedemonstreerd. Deze resultaten laten ook zien dat ze duidelijk beter presteren dan de onderhandelingsstrategieën uit de recente edities van de International Automated Negotiating Agents Competition (ANAC), en dit in een breed scala van onderhandelingsscenario's.

Onderzoeksvragen 3 en 4 worden in hoofdstukken 5 en 6 behandeld. Het momenteel meest gebruikte evaluatiecriterium voor onderhandelingsstrategieën is het behaalde prestatieniveau tijdens gesimuleerde toernooien. Dit criterium is helaas niet geschikt voor het evalueren van strategieën in dynamische scenario's, d.w.z. scenario's waar het de agenten toegelaten wordt om van strategie te wisselen tijdens de onderhandeling. In hoofdstuk 5 wordt getoond hoe empirische speltheorie (EGT) gebruikt kan worden voor het evalueren van strategieën in dit soort scenario's. Voor grootschalige evaluaties, waar een groot aantal agenten in deelnemen en ieder agent maar met een beperkt aantal agenten kan onderhandelen, zijn zowel de toernooi gebaseerde evaluatie alsook de EGT-evaluatie ongeschikt. In hoofdstuk 6 wordt getoond hoe ruimtelijke evolutionaire speltheorie gebruikt kan worden om de prestatie van strategieën in dit soort scenario's te meten.

Tot slot worden in hoofdstuk 7 de bijdragen van dit proefschrift samengevat en de door dit onderzoek mogelijk gemaakte richtingen voor toekomstig onderzoek geïdentificeerd.

Kortom, het in dit proefschrift omschreven onderzoek heeft de stand van zaken

in het veld van geautomatiseerde onderhandelingen op twee belangrijke manieren geavanceerd. Ten eerste, door het ontwikkelen van nieuwe onderhandelsstrategieën welke zeer effectieve technieken bezitten voor het modelleren van tegenstanders en het sluiten van concessies. Ten tweede, door te demonstreren hoe concepten en technieken uit de speltheorie gebruikt kunnen worden voor het evalueren van strategieën in praktijk relevante onderhandelingsscenario's, voor welke de traditionele en gebruikelijke toernooi-gebaseerde evaluatie ontoereikend is.

Curriculum Vitae

Siqi Chen was born on March 20, 1983 in Chongqing, China. He graduated from the University of Electronic Science and Technology of China with a BSc in Computer Science, and then started in September 2010 at the Department of Knowledge Engineering, Faculty of Humanities and Sciences of Maastricht University, with a combined MSc (Artificial Intelligence, 2-year programme) and Ph.D. study. His study was jointly funded by China Scholarship Council and Maastricht University. His scientific contributions resulted in various high-level international journal and conference publications such as ECAI, AAMAS, IJCAI, EAAI and ACM TAAS. He also has been invited to serve as a reviewer for a number of journals and conferences (e.g., BNAIC, PRIMA, Journal of ZUSC, JAAMAS, ACM TAAS and ESWA, etc). In 2012 he participated successfully in the International Automated Negotiating Agents Competition (ANAC) and achieved the third place.

Publication List

Publications 2014:

H.B. Ammar, S. Chen, K. Tuyls and G. Weiss. Automated transfer for reinforcement learning tasks. *Künstliche Intelligenz*, Vol. 28, Issue 1, 7-14, 2014.

S. Chen, H.B. Ammar, K. Tuyls and G. Weiss. Transfer for automated negotiation. *Künstliche Intelligenz*, Vol. 28, Issue 1, 21-27, 2014.

S. Chen and G. Weiss. An Approach to Complex Agent-based Negotiations via Effectively Modeling Unknown Opponents. (Under review)

S. Chen and G. Weiss. Using evolutionary game theory to evaluate complex agent-based negotiation strategies. (Under review)

S. Chen and G. Weiss. OMAC: A discrete wavelet transformation based negotiation agent. In I. Marsa-Maestre, M.A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai and K. Fujita (Eds), *Novel Insights in Agent-based Complex Automated Negotiation*. Springer-Verlag (Studies in Computational Intelligence, Vol. 535, pp. 187-196). 2014.

S. Chen and G. Weiss. An Intelligent Agent for Bilateral Negotiation with Unknown Opponents in Continuous-time Domains. *ACM Transactions on Autonomous and Adaptive Systems*, 9(3). 2014. (In press)

S. Zhou, S. Chen, E. Smirnov, G. Weiss and K. Tuyls. Automatic knowledge transfer for agent-based bilateral negotiation tasks. *Seventh International Workshop on Agent-based Complex Automated Negotiations (ACAN in conjunction with AAMAS)*. 2014.

Publications 2013:

S. Chen, H. B. Ammar, K. Driessens, K. Tuyls, and G. Weiss, “Automated Transfer Between Negotiation Tasks,” In *Proceedings of the Adaptive Learning Agents (ALA) at the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Minnesota, USA, 2013.

S. Chen, H. B. Ammar, K. Tuyls, and G. Weiss, “Optimizing Complex Automated Negotiations using Sparse Pseudo-Input Gaussian Processes”, In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 707-714), Minnesota, USA, 2013.

S. Chen, H. B. Ammar, K. Tuyls, and G. Weiss, “Using Conditional Restricted Boltzmann Machines for Highly Competitive Negotiation Tasks,” In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 69-75), Beijing, China, 2013.

S. Chen, H. B. Ammar, K. Tuyls and G. Weiss. Automated negotiation based on sparse pseudo-input Gaussian processes. In Proceedings of the 25th Benelux Conference on Artificial Intelligence (BNAIC, pp. 307-308), 2013.

S. Chen and G. Weiss. An efficient automated negotiation strategy for complex environments. *Engineering Applications of Artificial Intelligence* 26, 10(2013), 2613-2623. 2013.

Publications 2012:

S. Chen and G. Weiss. An efficient and adaptive approach to negotiation in complex environments. In Proceedings of the 20th European Conference on Artificial Intelligence (ECAI, pp. 228-233). 2012. IOS Press.

S. Chen and G. Weiss. A novel strategy for efficient negotiation in complex environments. In Proceedings of the 10th German Conference on Multiagent System Technologies (MATES, pp. 68-82). *Lecture Notes in Computer Science*, Vol. 7598. 2012. Best Paper Award. Springer-Verlag.

S. Chen, H. B. Ammar, K. Tuyls, and G. Weiss, "Transfer Learning for Bilateral Multi Issue Negotiations," In Proceedings of the Benelux Conference on Artificial Intelligence (BNAIC, pp. 59-66), Maastricht, The Netherlands, 2012.