Doctoral Dissertation

Automated Multi-Agent Negotiation Protocols for Highly Nonlinear Utility Space

Katsuhide Fujita

Department of Computer Science and Engineering

Nagoya Institute of Technology

June, 2011

Abstract

Negotiation is an important aspect of daily life and represents an important topic in the field of multi-agent systems research. While there has been a lot of previous work in this area, these efforts have, to date, dealt almost exclusively with simple negotiations involving independent multiple issues, and therefore, linear (single optimum) utility functions. Many real-world negotiation problems, however, involve interdependent multiple issues. The key impact of such issue dependencies is that they result in agent utility functions that are highly nonlinear, i.e. that have multiple optima. Negotiation mechanisms that are well suited for linear utility functions, unfortunately, fare poorly when applied to nonlinear problems.

In the multi-issue negotiation field, there are two main issues. First, the existing works aren't concerned with revealing the utility information of agents (Privacy). Excessively revealing utility information is not good for reaching agreements. Additionally, scalability for the complexity of the agent's utility function isn't very high (Scalability). A negotiation protocol should find high-quality solutions when the complexity of agents' utility function is high. In addition, most negotiation protocols are evaluated based on one's own testbed. For example, some previous negotiation protocols are only evaluated on randomly generated utility spaces. However, the effectiveness of the negotiation protocols is evaluated based on the same testbed. A common testbed based on negotiation in real life is necessary for generating effective negotiation protocols.

First, this thesis proposes a new threshold-adjusting mechanism in which agents who open their local information more than the others

can persuade the others. The experimental results demonstrate that the threshold-adjusting mechanism can reduce the amount of private information that is required for an agreement among agents. In addition, the experimental results show that the threshold-adjusting mechanism can reduce the computational cost while preserving enough optimality.

Next, this thesis proposes a representative-based protocol that has high scalability for the number of agents and considers the agent's private information. In our protocol, we first select representatives who revealed more of their utility space than the others. These representatives reached an agreement on alternatives and proposed them to the other agents. Finally, the other agents can express their own intentions concerning agreement or disagreement. In this protocol, agents who revealed more private utility information can have a greater chance to be representatives who will reach an agreement on behalf of the other agents. Although agents tend to avoid revealing their own private information, they have an incentive to reveal it in order to be representatives.

Third, this thesis proposes Distributed Mediator Protocol (DMP) and Take it or Leave it (TOL) Protocol for negotiation, which can reach agreements and completely conceal agents' private information. Moreover, this thesis proposes Hybrid Secure Protocol (HSP), which combines Distributed Mediator Protocol with Take it or Leave it Protocol. HSP can also reach agreements while completely concealing agents' private information. Furthermore, HSP achieves high optimality and incurs less communication cost. We demonstrate the performance of HSP in cone-constraint and cube-constraint situations.

Fourth, this thesis examines the problem that, in such domains, agent utility functions are nonlinear, and thereby can create nonconvex Pareto frontiers. This in turn implies that the Nash Bargaining Solution, which has been viewed as the gold standard for identifying a unique optimal negotiation outcome, does *not* serve that role in nonlinear domains. In nonlinear domains, unlike linear ones, there can be multiple Nash Bargaining Solutions, and all can be sub-optimal with respect to social welfare and fairness. This thesis proposes a novel negotiation protocol called the Secure and Fair Mediator Protocol (SFMP) that addresses this challenge, enabling secure multilateral negotiations with fair and Pareto-optimal outcomes in nonlinear domains. The protocol works by (1) using nonlinear optimization, combined with a Multi-Party protocol, to find the Pareto front without revealing an agent's private utility information, and (2) selecting the agreement from the Pareto set that maximizes a fair division criterion we call approximated fairness. We demonstrate that SFMP is able to find agreements that maximize fairness and social welfare in nonlinear domains, and that it out-performs (in terms of outcomes and scalability) previously developed nonlinear negotiation protocols.

Fifth, this thesis proposes another reasonable approach to reducing computational cost while maintaining good quality outcomes, which is to decompose the utility space into several largely independent subspaces based on four types of issue inter-dependencies. This method allows good outcomes with greater scalability than previous efforts. We also analyze how the types of issue interdependency influence the solution optimality and failure rate.

Finally, this thesis proposes a common testbed-generation tool based on XML that mainly covers the utility functions based on the constraints. First, we propose a testbed-generation tool that inputs configuration data and outputs XML-formatted files that represent agent utility spaces. The current tool can produce four types of utility spaces: Random, A Single Hill, Two Hills, and Several Hills. These types are observed in real negotiation settings. Also, we define the agent's utility space information based on XML formats. By defining the testbed data as XMLs, users can easily read the files and change the data structure.

論文要旨

交渉問題を扱う研究領域において，複数論点交渉問題が注目を集めている．筆者らは特に一般性が高く実世界に近い問題である複数の論点同士が相互依存関係にある交渉問題に注目している．例えば，自動車を購入に関する交渉問題に関して「車の大きさが大きければ多少値段が高くても購入する」などのように論点同士が相互依存関係の場合は現実的な例として多数存在する．多くの既存研究では論点の独立性が仮定されており非線形な効用関数に対し適用が困難である．

本論文では複数論点交渉問題において特に重要な問題とされているプライバシー情報の公開とプロトコルのスケーラビリティの 2 点に着目する．エージェントのプライバシー情報に関して，交渉の際にエージェントの効用情報が過剰に公開されるのは好ましくなく，できるかぎり非公開にする方がより現実的な設定である．また，交渉プロトコルのスケーラビリティが低い場合．複雑な交渉問題に対して良質な合意案を発見できない．本論文では以上の二点に注目した交渉プロトコルを提案する．

本論文では，まず，各エージェントがどれくらい自分の効用情報を公開しているかを表す指標として公開範囲を定義し，公開範囲に基づいて閾値の調整を行うメカニズムを提案する．さらに，閾値調整メカニズムと計算量の関係について議論し，本手法を用いることで各エージェントが過剰に公開することを防いでいることをシミュレーション実験により示す．

次に，交渉プロトコルのスケーラビリティに関して，代表エージェントという組み合わせ最適解を求めるステップに参加できるエージェントを定義し，計算量を削減する手法（Representative based Protocol）を提案する．さらに，本手法がエージェント数に対してスケーラブル

であり，合意形成失敗率を減少させていることをシミュレーション実験により示す．

メディエータを含めた他者に各エージェントの効用値を知られることなく合意形成が可能な分散メディエータに基づく交渉プロトコルを提案する．分散メディエータに基づく交渉プロトコルは暗号分野のセキュアマルチパーティプロトコルを導入させた手法である．また，分散メディエータに基づく交渉プロトコルの欠点であるエージェントとメディエータの通信量の増大を防ぐことに成功したハイブリッド型セキュア交渉プロトコルを提案する．

また，論点間の相互依存関係に基づき論点グループに交渉問題を再構成することでスケーラビリティを向上することができる．そこで，本論文ではエージェント間の相互依存関係に着目した論点グループに基づく交渉手法を提案する．論点グループに基づく交渉プロトコルは，メディエータが存在する相互依存度が最大になるように，全論点をグループごとに分割し，分割したグループごとに合意形成を行なう手法である．以上の手法に関してシミュレーション実験を行い，既存の手法と比較してスケーラビリティや最適性に関して比較を行う．

本論文で扱う制約を基にした効用関数では，効用関数の非線形性のために非凸性をもつことが分かっている．したがって，非線形性な効用空間を定義した場合，ナッシュ交渉解が単一にならない．さらに，ナッシュ交渉解が社会的効用最大かつ均等解である保証がない．そこで，本論文では近似公平性の概念を導入したセキュアかつ公平な交渉手法 (SFMP) を提案する．本プロトコルは，まず，エージェントの効用情報を公開することなく，パレートフロントを探索する．その後．近似公平性 (Approximated Fairness) に基づいて，合意案の公平性の観点から最終的な合意案を決定する．シミュレーション実験から，社会的効用最大かつ公平な合意案を得ることが困難な非線形効用関数の場合でも，社会的効用最大かつ公平性を考慮した合意案を発見可能となることを示す．

各論点が相互依存関係の場合の交渉問題を対象としたプロトコルは，共通のテストベッドが存在せず，個々のテストベッドによって評価されている．特に多くの既存研究ではランダムに生成された効用関数の

場合しか評価されていない．交渉手法の有効性を評価するためには共通したテストベッドを用いて，交渉手法を様々な角度から検証される必要がある．本論文では各論点が相互依存関係の場合に対応した交渉プロトコルを評価するため，XMLを基にした共通テストベッドを提案する．また，共通テストベッドを利用した交渉プロトコル評価プログラムのオープンソース化に関して述べる．

Acknowledgements

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, only some of whom it is possible to mention here.

This thesis would not have been possible without the help, support and patience of my principal supervisor, Prof. Takayuki Ito at the Nagoya Institute of Technology. I want to thank him for his constant support and encouragement in all aspects of my research, career and even personal issues. He is always enthusiastic in sharing his experience and knowledge with me. I am grateful for the opportunity to work with and to learn from such an extraordinary supervisor.

I also especially want to thank Dr. Mark Klein at the Massachusetts Institute of Technology. He always gives me precise and kind advice for my study. In addition, he gave me a great chance to stay at MIT as a visiting student. This experience has been of great value to my future career and research life.

I would like to acknowledge the financial, academic and technical support of the all Professors and staff at the Nagoya Institute of Technology. In particular, I would like to thank Prof. Nobuhiro Inuzuka and Prof. Hiroshi Matsuo for being my thesis judges. This thesis would not have been possible without their advice and contributions. Also, I would like to acknowledge the researchers of the field of the Multi-Agent Systems. In particular, Prof. Makoto Yokoo at Kyushu University gave me helpful comments as a researcher representing the field of the Multi-Agent Systems.

I gratefully thank all alumni, master's and undergraduate students in the Takayuki Ito Laboratory, Nagoya Institute of Technology. I had such a wonderful time with you in this laboratory through discussions, seminars, trips and parties. The events in this laboratory are cherished memories.

Above all, my parents and sister have given me their unequivocal support throughout, as always, for which my mere expression of thanks does not suffice.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background of Automated Negotiations

Multi-agent systems are one of the most promising technologies to emerge in recent decades at the crossroads between several fields such as artificial intelligence (AI), distributed systems, economics and even sociology. Many researchers have outlined a vision in which many of the tasks performed today by humans are delegated to intelligent, autonomous and proactive programs, generally called software agents. A system composed of several such agents is called a multi-agent system (MAS).

Negotiation is also an important aspect of daily life and represents an important topic. They can be simple and ordinary, as in haggling over a price in the market or deciding on a meeting time; or they can be complex and extraordinary, perhaps involving international disputes and nuclear disarmament [55] issues that affect the well-being of millions. While the ability to negotiate successfully is critical for much social interaction, the act of negotiation is not an easy task. Something that might be perceived as a "simple" case of a single-issue bilateral bargaining over a price in the marketplace can demonstrate the difficulties that arise during the negotiation process.

It is a subject that has been extensively discussed in game-theoretic, economic, and management science literature for decades (e.g. [17, 23, 24, 33, 103, 113, 121, 134]). Recently, more and more researchers in the multi-agent field are interested in automated negotiation systems that consist of intelligent software agents[57, 59, 60, 61, 72, 88, 146]. In fact, there has been extensive work in the area of automated negotiation; that is, where automated agents negotiate with other agents in such contexts as e-commerce [77, 114, 124, 126], large scale argumentation [75, 96], collaborative design [63, 64], and service-oriented computing [11, 76]. The model of the multi-agent system is necessary for cooperative work between agents, and automated negotiations between software agents are required when they work together. In addition, most researchers in multi-agent systems regard automated negotiation as the most important topic for theoretical analysis or practical applications of agent-based systems.

The many benefits of such agents include alleviating some of the efforts required of humans during negotiations and assisting individuals who are less qualified in the negotiation process, or in some situations, replacing human negotiators altogether [46, 82, 85]. Another possibility is for people embarking on important negotiation tasks to use these agents as a training tool, prior to actually performing the task [51]. Thus, success in developing an automated agent with negotiation capabilities has great advantages and implications.

1.1.1 Main Issues of Automated Negotiations

The main issues of accomplishing the automated negotiations are as follows: Negotiation Environment, Preference Elicitation, Automated Negotiation Protocol.

Negotiation Environment The negotiation environment defines the specific settings of the negotiation. Based on these settings, different considerations should then be taken into account. The environment determines several parameters that dictate the number of negotiators taking part in the negotiation, the time frame of the negotiation, and the issues on which the negotiation is being conducted. The number of parties participating in the negotiation can be two (bilateral negotiations) or more (multilateral negotiations). The negotiation environment also consists of a set of objectives and issues to be resolved. Various types of issues can be involved, including discrete enumerated value sets, integer-value sets, and real-value sets. Negotiations that involve multi-attribute issues allow making complex decisions while taking into account multiple factors [69].

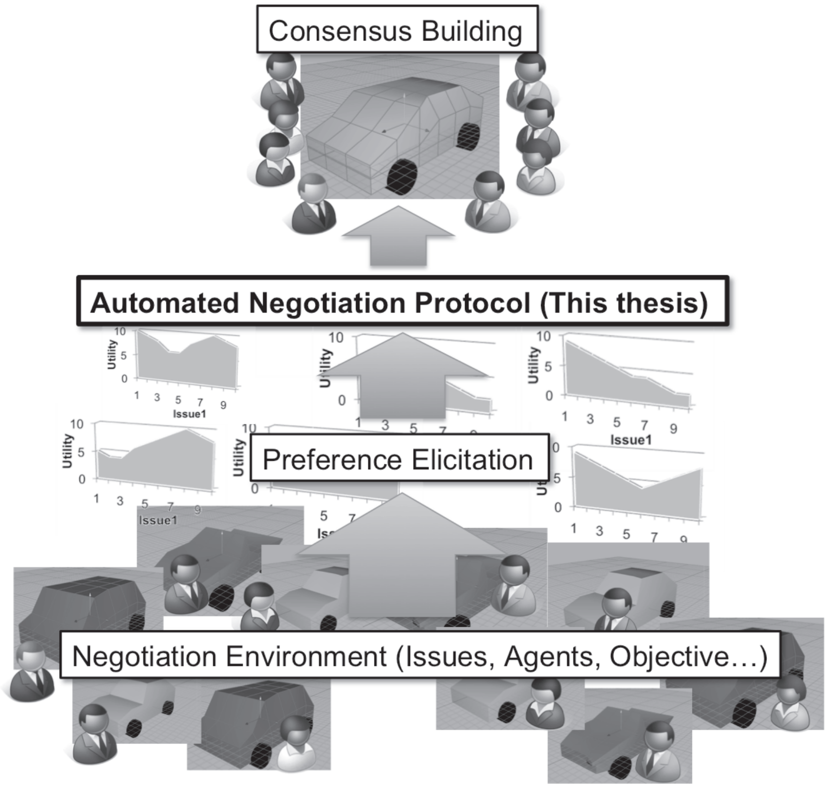

Figure 1.1: Issues of Automated Negotiations

Preference Elicitation Preference elicitation techniques attempt to collect as much information on users' preferences as possible in order to find efficient solutions [92, 125]. Because users' preferences are always incomplete initially and tend to change in different contexts, in addition to user's cognitive and emotional limitations of information processing, preference elicitation methods must also be able to avoid preference reversals, discover hidden preferences, and assist users in making tradeoffs when confronted with competing objectives.

Automated Negotiation Protocol Automated negotiation protocol defines the formal interaction between the decision makers (Agents) in the negotiation environments -whether the negotiation is done only once (one-shot) or repeatedly- and how the exchange of offers between the agents is conducted.

> In addition, according to Jennings et al. [62], a negotiation protocol is a set of rules that govern the interaction and cover the permissible types of participants (e.g., the negotiators and any relevant third parties), the negotiation states (e.g., accepting bids, negotiation closed), the events that cause negotiation states to change (e.g., no more bidders, bid accepted) and the valid actions of the participants in particular states (e.g., which messages can be sent by whom, to whom, at what stage).
>
> The agents in the negotiations can be non-cooperative or cooperative. Generally, cooperative agents try to maximize their combined joint utilities (see Zhang [157]) while non-cooperative agents try to maximize their own utilities regardless of the other side's utilities. This thesis focuses on these kinds of issues, which have been widely studied in different research areas, such as game theory [103, 121], distributed artificial intelligence [27, 29, 74] and economics [113].

Figure 1.1 shows the concept of the main issues in accomplishing automated negotiations. This figure shows the example of designing a simple car among car designers. First, the negotiation environment, including negotiation issues, agents' actions, and objectives, is defined based on negotiation in real life. Next, the preference information of the users' should be collected using some preference elicitation techniques. Third, agents negotiate the car designs automatically based on the negotiation protocol. One of the most important parts of automated negotiation is the negotiation protocol, and it is a subject that has been extensively discussed in game-theoretic, economic, and management science literature for decades. In addition, there are many problems that remain to be solved in the negotiation protocol, and these problems constitute the main research theme in the multi-agent system field. This thesis focuses on the automated negotiation protocol in order to accomplish automated negotiations. Finally, agents build a consensus for designing the car.

1.1.2 Complex Multi-Issue Negotiation with Highly Non-linear Utility Functions

In this thesis, the automated negotiation protocols between cooperative agents are focused on. While there has been a lot of previous work in this area [30, 80, 135], these efforts have, to date, dealt almost exclusively with simple negotiations involving independent multiple issues, and therefore, linear (single optimum) utility functions. An example of such representations widely used in the negotiation literature is linear-additive utility functions [29], which allow modeling of independent issues.

Many real-world negotiation problems, however, involve interdependent multiple issues that are highly nonlinear. When designers work together to design a car, for example, the value of a given carburetor is highly dependent on which engine is chosen. The addition of such interdependencies greatly complicates the agent's utility functions, making them nonlinear, with multiple optima. Interdependence between attributes in agent preferences can be described by using different categories of functions, like K-additive utility functions [15, 43], bidding languages [10] or constraints [56, 79, 93].

In fact, in the context of a multi-attribute negotiation, complexity depends on the number of issues, the number of agents, the level of interdependency between the preferences on the issues and the domain of the issues. The method to describe the agent's utility spaces also represents a fundamental measure in the complexity of the negotiation scenario.

Recently, some studies have focused on negotiation with nonlinear utility functions. Klein et al. [74] present the first negotiation protocols specifically for complex preference spaces. They focus on the nonlinear utility function, and describe a simulated annealing-based approach appropriate for negotiating complex contracts that achieves near-optimal social welfare for negotiations with binary issue dependencies. The important points in this work are the positive results regarding the use of simulated annealing as a way to regulate agent decision making, along with the use of agent expressiveness to allow the mediator to improve its proposals. In addition, most existing negotiation protocols like a method based on Hill-climbing, which is well-suited for linear utility functions, work poorly when applied to nonlinear problems. The contribution of this paper is significant, however, it was not applied to multilateral negotiations with higher-order dependencies. Higher-order dependencies and continuous-valued issues, common in many real-world contexts, are known to generate more challenging utility landscapes that are not considered in their work.

One of the most relevant approaches focusing on the complex utility space is Ito et al. [56, 58]. The important contributions of this paper are the original constraint-based utility functions, which assume highly nonlinear and bumpy utility functions. Therefore, scalable and efficient negotiation protocols are required if the complexity of the negotiation environment is high. Also, this paper proposes a bidding-based protocol. In this protocol, agents generate bids by sampling their own utility functions to find local optima, and then use constraint-based bids to compactly describe regions that have large utility values for that agent. A mediator then finds a combination of bids that maximizes social welfare. This protocol also had an impact on the automated negotiation field because many existing works didn't consider the highly nonlinear utility of agents.

In this thesis, the constraint-based nonlinear utility function is focused on. There are many multi-issue negotiation models except the use of constraints; however, there are several reasons in favor of using constraints in negotiation models. First, they allow for efficient methods of preference elicitation. Moreover, constraints allow expression of dependencies between the possible values of the different attributes. Finally, the use of constraints for offer expression allows limiting of the region of the solution space that has to be explored in a given negotiation step. Reducing the region of the utility space under exploration according to the constraints exchanged by agents is a widely used technique in automated negotiation [49, 91], since it makes the search for agreements a more efficient process than when using positional bargaining, especially in complex negotiation scenarios.

1.2 Main Contributions of this thesis

Focusing on the complex multi-issue automated negotiation protocol, there are some unsolved issues in the existing works. This thesis deals with the followings.

Aim 1: Scalable and Efficient Negotiation Protocols

An important problem is **scalability** for the number of agents and the number of issues. In our negotiation setting, the utility space becomes extremely nonlinear, making it very difficult to find the optimal agreement point. For example, the bidding-based negotiation protocol does not have high scalability for the number of agents, and the mediator needs to find the optimum combination of submitted bids from the agents. However, the computational complexity for finding solutions is too large.

This thesis proposes a representative-based protocol that has high scalability for the number of agents and considers the agent's private information (Chapter 5). In our protocol, the mediator first selects representatives who revealed more of their utility space than the others. These representatives reached an agreement on alternatives and proposed them to the other agents. Finally, the other agents can express their own intentions concerning agreement or disagreement. In this protocol, agents who revealed more private utility information can have a greater chance to be representatives who will reach an agreement on behalf of the other agents. Although agents tend to avoid revealing their own private information, they have an incentive to reveal it in order to be representatives. The representative-based protocol was inspired by the parliamentary systems in England, Canada, Australia, Japan, etc. in which representatives are making an agreement on behalf of other people. In a situation in which many people have to reach an agreement, directly reflecting all members' opinions is quite difficult. Doing so requires much time and energy and is not scalable. Although voting is one option, voting might have paradoxical results [4]. In addition, our mechanism is expandable to be multi-round by using the Threshold Adjustment Protocol [35]. The multi-round mechanism improves the failure rates and achieves fairness in terms of the revealed area.

This thesis also proposes another protocol based on decomposing the contract space based on issue interdependencies (Chapter 8). A new protocol in which a mediator tries to reorganize a highly complex utility space into several tractable utility subspaces is proposed in order to reduce the computational cost. Issue groupings are generated by a mediator based on an examination of the issue interdependencies. First, a measure for the degree of interdependency between issues is defined. In this thesis, four such measures are defined. Second, a weighted non-directed interdependency graph is generated based on this information. By analyzing the interdependency graph, a mediator can identify issue subgroups. Note that while others have discussed issue interdependencies in utility theory [25, 26, 140], this previous work doesn't identify optimal issue groups. Finally, this thesis demonstrates that our protocol, based on issue-groups, has higher scalability than previous efforts, and discusses the impact on the optimality of the negotiation outcomes.

Aim 2: Negotiation Protocols Concerning Agents' Private Information

First, the existing works have not yet been concerned about agents' private information (**privacy**). Such private information should be protected as much as possible in negotiation because users generally value privacy in real life. For example, suppose that several companies collaboratively design and develop a new car model. If one company reveals more private information than the other companies, the other companies will know more of that company's important information, such as utility information. As a result, the company will be at a disadvantage in subsequent negotiations, and the mediator might leak the agent's utility information. Therefore, the aim of this thesis is to accomplish the negotiation protocols without revealing the agents' private information to others.

This thesis proposes a threshold-adjusting mechanism (Chapter 4). First, agents make bids that produce more utility than the common threshold value based on the bidding-based protocol proposed in [56]. Then, the mediator asks each agent to reduce its threshold based on how much each agent opens its private information to the others. Each agent makes bids again above the threshold. This process continues iteratively until agreement is reached or there is no solution. Our experimental results show that our method substantially outperforms the existing negotiation methods on the point of how much agents have to open their own utility space.

In addition, this thesis proposes secure protocols to conceal all agent private information: the Distributed Mediator Protocol (DMP) and the Take it or Leave it (TOL) Protocol (Chapter 6). They make agreements and conceal agent utility values. When searching in their search space, they employ Secure Gathering, with which they can simultaneously calculate the sum of the per agent utility value and conceal it. Furthermore, Distributed Mediator Protocol (DMP) improves the scalability for the complexity of the utility space by dividing the search space toward the mediators. In the Take it or Leave it (TOL) Protocol, the mediator searches using the hill-climbing search algorithm. The evaluation value is decided by responses that agents either take or leave moving from the current state to the neighbor state. The Hybrid Secure Protocol (HSP) that combines DMP with TOL is proposed. In HSP, TOL is performed first to improve the initial state in the DMP step. Next, DMP is performed to find the local optima in the neighborhood. HSP can also reach an agreement and conceal per-agent utility information. Additionally, HSP can reduce the required memory for making an agreement, which is a major issue in DMP. Moreover, this thesis demonstrates that HSP can improve communication cost (memory usage) more than DMP.

Aim 3: Addressing Weaknesses of the Nash Bargaining Solution in Nonlinear Negotiation

The Nash bargaining solution, which maximizes the product of the agent utilities, is a well-known metric that provably identifies the optimal (fair and social-welfare-maximizing) agreement for negotiations in linear domains [68, 71, 103]. In *nonlinear* domains, however, the Pareto frontier will often not satisfy the convexity assumption required to make the Nash solution optimal and unique [21, 68, 103]. There can, in other words, be multiple agreements in nonlinear domains that satisfy the Nash Bargaining Solution, and many or all of these will have sub-optimal fairness and/or social welfare.

This thesis proposes a secure mediated protocol (the Secure and Fair Mediator Protocol, or SFMP) that addresses this challenge (Chapter 7). The protocol consists of two main steps. In the first step, SFMP uses a nonlinear optimizer, integrated with a secure information sharing technique called Secure Gathering [131], to find the Pareto front without causing agents to reveal private utility information. In the second step, an agreement is selected from the set of Pareto-optimal contracts using a metric called approximate fairness that measures how equally the total utility is divided across the negotiating agents ([115] etc.) This thesis demonstrates that SFMP produces better scalability and social welfare values than previous nonlinear negotiation protocols.

Aim 4: Common Testbed for Multiple Interdependent Issue Negotiation Problems

Most negotiation protocols are evaluated based on one's own testbed. For example, some works [49, 56] were only evaluated on randomly generated utility spaces. However, the effectiveness of the negotiation protocols is evaluated based on the same testbed. Thus, this thesis proposes a tool that generates testbeds for evaluating multi-issue negotiation protocols by focusing on the utility function based on constraints.

A common testbed-generation tool based on XML is proposed (Chapter 9). The input is the configuration files that define the number of issues, the number of agents, etc. The testbed-generation tool produces XML files that define the agent's utility spaces in XML format as output. This tool has four types of utility spaces: Random, Single Hill, Two Hills, and Several Hills, which are based on actual negotiation settings. This thesis defines XML tags, which represent utility spaces, that consist of cone-based and cube-based constraints. By utilizing an XML format, users can easily understand, modify, and update the meaning of the data and exchange the data among research communities. In addition, our XML format does not depend on a certain environment. This thesis shows cube-based and cone-based constraint tags that define the building blocks of utility function spaces.

1.3 Thesis Organization

Figure 1.2 shows the organization of the thesis, the remainder of which is organized as follows. First, this thesis describes related works for automated negotiation fields (Chapter 2). Second, it presents a model of nonlinear negotiation with utility functions based on constraints (Chapter 3). Third, this thesis proposes a threshold-adjusting mechanism focusing on the privacy issue (Chapter

4). Fourth, this thesis proposes a representative-based protocol that has high scalability for the number of agents and considers the agent's private information (Chapter 5). Fifth, this thesis proposes secure protocols to conceal all agent private information. In particular, the Distributed Mediator Protocol (DMP) and the Take it or Leave it (TOL) Protocol are proposed (Chapter 6). Sixth, a new protocol (SFMP) designed to address the Nash bargaining solution challenge in nonlinear situations is described (Chapter 7). Seventh, another protocol based on decomposing the contract space based on issue interdependencies is proposed (Chapter 8). Eighth, a common testbed for evaluating the multi-issue negotiations is proposed. Finally, I draw the conclusions of this thesis.

1.4 Publications related to each chapter

All the chapters of this thesis are based on reviewed journals, as follows:

- **Chapter 3 and Chapter 4:** Katsuhide Fujita, Takayuki Ito, Hiromitsu Hattori, "Using Autonomous Threshold Adjustment to Enable Multi-Party Negotiations with Multiple Interdependent Issues," Computer Software (The Journal of JSSST), Japan Society for Software Science and Technology (JSSST), Vol.25, No.4, pp.167-180, 2008 and Katsuhide Fujita, Takayuki Ito, "An Analysis of Computational Complexity of the Threshold Adjusting Mechanism in Multi-Issue Negotiations", International Transactions on Systems Science and Applications, SIWN, Vol.4, No.4, pp.305-311, 2008

- **Chapter 5:** Katsuhide Fujita, Takayuki Ito, Mark Klein, "Representative-based Multi-Round Protocol for Multiple Interdependent Issues Negotiations", Multiagent and Grid Systems, Vol.6, No.5-6, pp.459-476, 2010.

- **Chapter 6:** Katsuhide Fujita, Takayuki Ito, Mark Klein, "Secure and Efficient Protocols for Multiple Interdependent Issues Negotiation," Journal of Intelligent and Fuzzy Systems 21(3): pp.175-185, 2010 and Katsuhide Fujita, Takayuki Ito,and Mark Klein, "Secure and Scalable Protocols for Multiple Interdependent Issues Negotiations", IEEJ Transaction on Electronics, Information and Systems (Sec.C), Vol.130, No.4 , pp.651-659, 2010.

- **Chapter 7:** Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Secure and Fair Protocol that Addresses Weaknesses of the Nash Bargaining Solution in Nonlinear Negotiation," Group Decision and Negotiation, Springer, DOI: 10.1007/s10726-010-9194-6, 2010.

- **Chapter 8:** Katsuhide Fujita, Takayuki Ito, Mark Klein, "Scalable and Efficient Protocols by Grouping Issues in Multiple Interdependent-Issue Negotiations," Japan Society of Artificial Intelligence (JSAI), Vol.26, No.1, pp.147-155, 2011 and Katsuhide Fujita, Takayuki Ito, Mark Klein, "An Automated Consensus Mechanism based on Adjustment of Issue-Groups for Multi-issue Negotiation," Journal of Information Processing Society of Japan , Vol.52, No.4, 2011.

- **Chapter 9:** Katsuhide Fujita, Takayuki Ito, Mark Klein, "Common Testbed Generating Tool based on XML for Multiple Interdependent Issues Negotiation Problems," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.15 No.1, pp. 34-40, 2011.

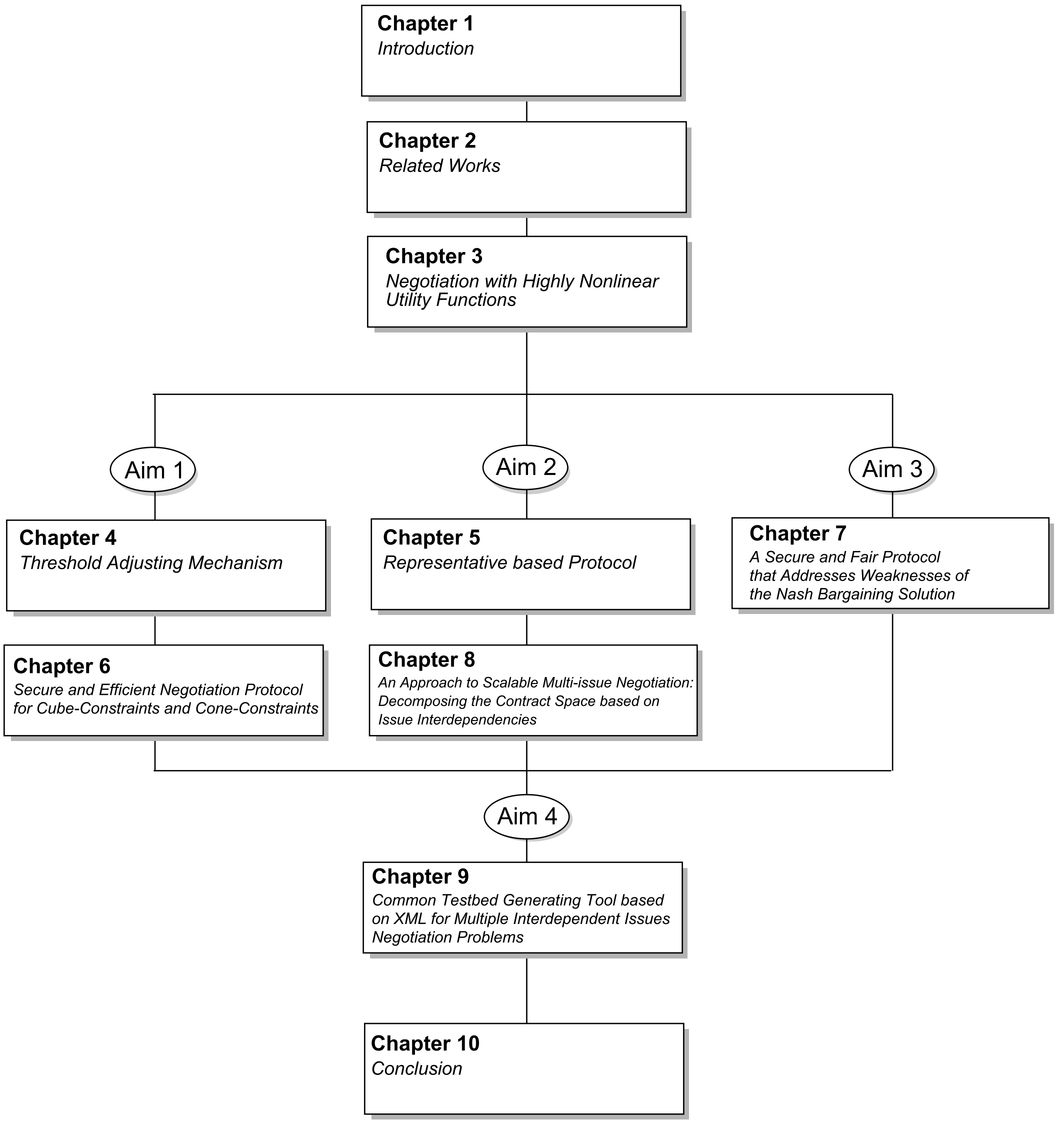

Figure 1.2: Thesis Organization

2

Related Works

2.1 Introduction

This chapter shows the extensive related works of this thesis including bargaining theory, auction mechanism, optimization techniques, and recent studies of multi-issue automated negotiations. In 2.2, we describe the multi-agent systems, and 2.3 shows studies of the automated negotiation protocol. In 2.4, we describe the Contract Net Protocol. In 2.5, we illustrate the auction mechanism including combinational auctions. In 2.6, we describe the Distributed Constraint Optimization Problem. In 2.7, we discuss the recent related works in automated multi-issue negotiation, and in 2.8, we describe the negotiation competitions. Finally, we conclude this chapter.

2.2 Summary of Multi-agent systems

This thesis focuses on the negotiation in multi-agent systems. First, summaries of the agents and multi-agent systems are given. The automated intelligent agents are assumed as follows [150].

Intelligent Agents are:

1. clearly identifiable problem solving entities with well-defined boundaries and interfaces;

2. situated (embedded) in a particular environmentthey receive inputs related to the state of their environment through sensors and they act on the environment through effectors;

3. designed to fulfill a specific purposethey have particular objectives (goals) to achieve;

4. autonomousthey have control over both their internal state and their own behavior;

5. capable of exhibiting flexible problem-solving behavior in pursuit of their design objectivesthey need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and proactive (able to act in anticipation of future goals).

These days, a multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may include some methodic, functional, procedural or algorithmic search, find and process approach. In MAS, intelligent agents need to interact with one another, either to achieve their individual objectives or to manage the dependencies that follow from being situated in a common environment [12]. These interactions can vary from simple information interchanges, to requests for particular actions to be performed, and on to cooperation (working together to achieve a common objective) and coordination (arranging for related activities to be performed in a coherent manner).

One of the relevant interactions in MAS is *negotiation* (the process by which a group of agents come to a mutually acceptable agreement on some matter). Negotiation examines whether to cooperate and coordinate or not (both between artificial and human agents) and is required both when the agents are self-interested and when they are cooperative. In other words, negotiation is a significant method of competitive (or partially cooperative) allocation of goods, resources, or tasks between agents.

2.3 Automated Negotiation Protocols

The main issues in accomplishing automated negotiations are Negotiation Environment, Preference Elicitation, and Automated Negotiation Protocol. As described in Chapter 1, this thesis focuses on the automated negotiation protocol in order to accomplish automated negotiations. In focusing on the automated negotiation protocol, it should be mainly be considered to deal with three broad topics [89].

Negotiation Protocols: the set of rules that govern the interaction. This covers the permissible types of participants (e.g. the negotiators and any relevant third parties), the negotiation states (e.g. accepting bids, negotiation closed), the events that cause negotiation states to change (e.g. no more bidders, bid accepted) and the valid actions of the participants in particular states (e.g. which messages can be sent by whom, to whom, at what stage).

Negotiation Objects: the range of issues over which agreement must be reached. At one extreme, the object may contain a single issue (such as price), while on the other hand it may cover hundreds of issues (related to price, quality, timings, penalties, terms and conditions, etc.) Orthogonal to the agreement structure, and determined by the negotiation protocol, is the issue of the types of operation that can be performed on agreements. In the simplest case, the structure and the contents of the agreement are fixed and participants can either accept or reject it (i.e. a take it or leave it offer). At the next level, participants have the flexibility to change the values of the issues in the negotiation object (i.e. they can make counter-proposals to ensure the agreement better fits their negotiation objectives). Finally, participants might be allowed to dynamically alter (by adding or removing issues) the structure of the negotiation object (e.g. a car salesman may offer one year's free insurance in order to clinch the deal).

Agents' Decision-Making Models: The decision-making apparatus the participants employ to act in line with the negotiation protocol in order to achieve their objectives. The sophistication of the model, as well as the range of decisions that have to be made, are influenced by the protocol in

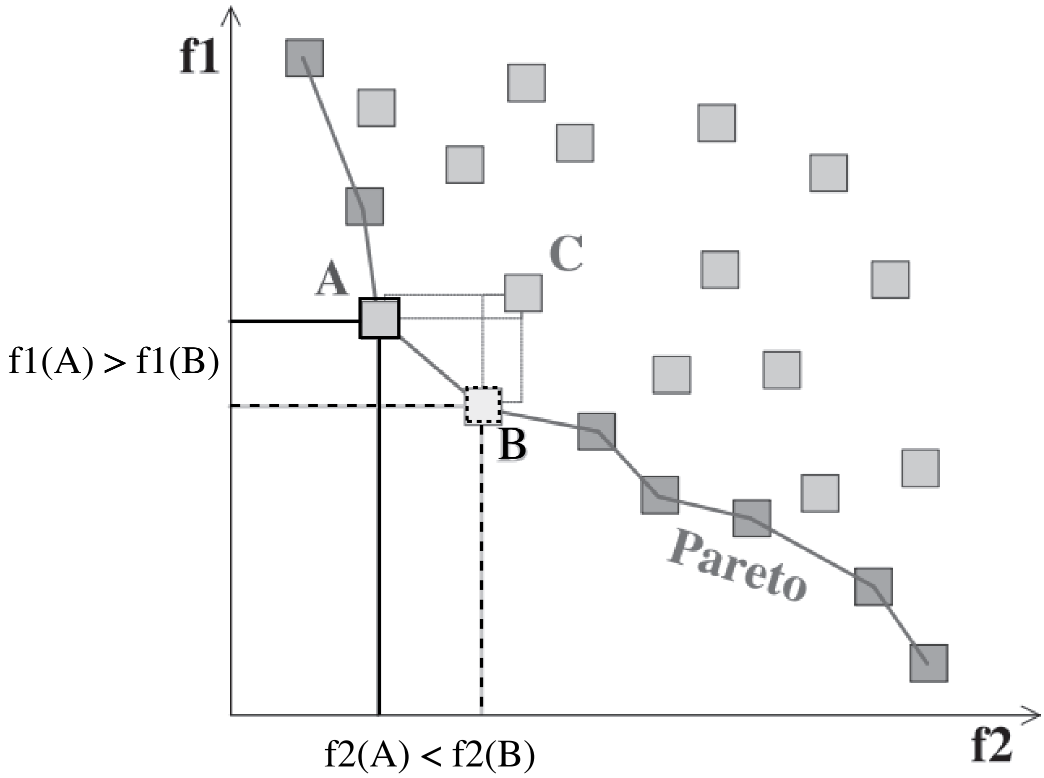

Figure 2.1: Example of Pareto frontier

place, by the nature of the negotiation object, and by the range of operations that can be performed on it.

The relative importance of these three topics varies according to the negotiation and environmental context. Therefore, several kinds of taxonomies of negotiation protocols are appearing now.

2.3.1 Properties of designing negotiation protocols

Protocols for governing multi-agent interactions are designed to have certain desirable properties [62]. Usually, these properties are based on the mechanism design field. Possible properties are described as follows.

Guaranteed success: A protocol guarantees success if it ensures that, eventually, agreement is certain to be reached.

Maximizing social welfare: Intuitively, a protocol maximizes social welfare if it ensures that any outcome maximizes the sum of the utilities of negotiation participants. If the utility of an outcome for an agent was simply defined in terms of the amount of money that agent received in the outcome, then a protocol that maximized social welfare would maximize the total amount of money "paid out."

Pareto efficiency [34, 106]: A negotiation outcome is said to be Pareto efficient if there is no other outcome that will make at least one agent better off without making at least one other agent worse off. Intuitively, if a negotiation outcome is not Pareto efficient, then there is another outcome that will make at least one agent happier while keeping everyone else at least as happy.

Given a set of choices and a way of valuing them, the Pareto frontier (Pareto set or Pareto front) is the set of choices that are Pareto efficient. The Pareto frontier is particularly useful in engineering: by restricting attention to the set of choices that are Pareto-efficient, a designer can make tradeoffs within this set, rather than considering the full range of every parameter.

Figure 2.1 shows an example of a Pareto frontier. The boxed points represent feasible choices, and smaller values are preferred to larger ones. Point C is not on the Pareto Frontier because it is dominated by both points A and B. Points A and B are not strictly dominated by any other, and hence do lie on the frontier.

Individual rationality: A protocol is said to be individually rational if following the protocol "playing by the rules" is in the best interests of negotiation participants. Individually rational protocols are essential because without them, there is no incentive for agents to engage in negotiations.

Stability: A protocol is stable if it provides all agents with an incentive to behave in a particular way. The best-known kind of stability is Nash equilibrium:

two strategies s and s' are said to be in Nash equilibrium if under the assumption that one agent is using s, the other can do no better than use s', and vice versa.

Simplicity: A "simple" protocol is one that makes the appropriate strategy for a negotiation participant "obvious." That is, a protocol is simple if, using it, a participant can easily (tractably) determine the optimal strategy.

Distribution: A protocol should ideally be designed to ensure that failure of consensus is low.

2.3.2 Classification of negotiation protocols

The classification of negotiation protocols, which takes effect in the negotiation situations, is as follows [116].

Number of negotiation parties: Many studies focus on the number of negotiation parties (agents) because the number of negotiation parties affects the complexity of finding agreements and agents' strategies. Usually, one-one (bilateral) negotiations, one-many negotiation (e.g. one-many bargaining, auctions) and many-many (e.g. contract net protocol) are popular taxonomies.

Number of issues under negotiation: Many studies focus on the number of issues under negotiation because it affects the complexity of finding agreements and agents' strategies. Single issue negotiation (e.g. only over price) and multi-issue negotiation are popular taxonomies.

Utility Functions: Utility function is a function that maps all possible game outcomes in the choice set into cardinal utility or ordinal utility. Cardinal utility is mapping to a real number (e.g. between $0 \sim 1$), and ordinal utility specifies only an (partial) ordering between outcomes [69, 142].

Complexity of preferences over the issues: The utility function is linearly additive utility functions or non-linear (e.g. k-additive). As described in Chapter 1, this thesis focuses on the complex utility functions that are nonlinear [74].

Integrative or Sequential "Integrative" (or "Global") negotiation protocol means that all issues are negotiated at the same time. On the other hand, "Sequential" protocol involves making step-by-step agreement one issue at a time. When the issues are independent, negotiation protocols take effect separately; however, all issues must be agreed upon before the agreement takes effect in the simultaneous implementation [16, 145].

Degree of self-interest on the part of the agents The relationship between agents is strictly competitive or cooperative. If agents are strictly competitive, it is hard for agents to make agreements.

Method of modeling time pressure In negotiation settings, time pressure is also an important topic. For example, fixed deadlines and time-discounting factors are popular [121].

Also, the negotiation protocol is usually generated by certain techniques in MAS, Game Theory and Logic. Another classification of negotiation protocols is based on the field of approach.

Game Theoretic Approach: This study assumes that rules of the game, preferences and beliefs of all players are common knowledge, and full rationality on the part of all players (unlimited computation). Preferences are encoded in a set of player types, closed systems, predetermined interaction, and small-sized games. However, the concepts of contracts and analysis are significant to negotiation protocol in nonclassical negotiation theory [50].

Heuristic Approach This approach is based on AI, MAS research. No common knowledge or uncertain rationality assumptions are needed. Agent behavior is modeled directly based on a negotiator in real life. Negotiation protocol based on the Heuristic Approach is suitable for open, dynamic environments, and the space of possibilities is very large.

Argumentation-based Approach This is a negotiation protocol based on formal logics of dialogue games. This field has lately been referred to simply as "Argumentation."

The details of the automated negotiation protocol based on the game theory, heuristic and argumentation are described as follows.

2.3.3 Game Theoretic Approaches

Game theory is a branch of economics that studies interactions between self-interested agents. Game theory is relevant to the study of automated negotiation because the participants in such negotiations can reasonably be assumed to be self-interested. Game theoretic studies of rational choice in multi-agent encounters typically assume that agents are allowed to select the best strategy from the space of all possible strategies, by considering all possible interactions. It turns out that the search space of strategies and interactions that needs to be considered has exponential growth, which means that the problem of finding an optimal strategy is in general computationally intractable. In computer science, the study of such problems is the domain of computational complexity theory.

In game theory, there are two distinct branches of research: Cooperative game theory and Non-cooperative game theory. Cooperative game theory abstracts away the specific rules of the game, and is concerned with finding solutions given a space of possible outcomes. In cooperative game theory, groups of players ("coalitions") may enforce cooperative behavior; hence the game is a competition between coalitions of players, rather than between individual players. Non-cooperative bargaining theory studies games with well-defined protocols and strategies, and focuses on the behavior of intelligent agents.

Bargaining is a type of negotiation in which the buyer and seller of a good or service dispute the price that will be paid and the exact nature of the transaction that will take place, and eventually come to an agreement. Bargaining is an alternative pricing strategy to fixed prices. Optimally, if it costs the retailer nothing to engage and allow bargaining, he can divine the buyer's willingness to spend. It allows for capturing more consumer surplus as it allows price discrimination, a process whereby a seller can charge a higher price to one buyer who is more eager (by being richer or more desperate). Negotiation and especially bargaining were studied in the game theory literature well before the emergence of multi-agent systems as a research discipline, and even before the advent of the first digital computer.

The important concept in bargaining theory is equilibrium. For example, strategies of all players are said to be in *Nash equilibrium* if no other party can benefit by unilaterally changing his/her strategy[104]. Other equilibrium concepts are *Dominant Equilibrium* and *Bayes-Nash Equilibrium. Dominant Equilibrium* occurs when one strategy is better than another strategy for one player, no matter how that player's opponents may play. The opposite, intransitivity, occurs in games where one strategy may be better or worse than another strategy for one player, depending on how the player's opponents may play. A *Bayesian Nash equilibrium* is defined as a strategy profile and beliefs specified for each player about the types of the other players that maximizes the expected payoff for each player given their beliefs about the other players' types and given the strategies played by the other players. In this section, therefore, we describe the non-cooperative games to determine whether the equilibrium is significant [107].

In a classical bargaining problem, the result is an agreement reached between all interested parties, or the status quo of the problem. It is clear that studying how individual parties make their decisions is insufficient for predicting what agreement will be reached. However, classical bargaining theory assumes that each participant in a bargaining process will choose between possible agreements, following the conduct predicted by the rational choice model. It is particularly assumed that each player's preferences regarding the possible agreements can be represented by a von Neumann-Morgenstern utility theorem function [105].

Nash [103] defines a classical bargaining problem as being a set of joint allocations of utility, some of which will correspond to what the players would obtain if they reach an agreement, and another that represents what they would get if they failed to do so. The Nash bargaining game [103] is a simple two-player game used to model bargaining interactions. In the Nash Bargaining Game, two players demand a portion of some good (usually some amount of money). If the two proposals sum to no more than the total good, then both players get their demand. Otherwise, both get nothing. A Nash bargaining solution is a Pareto-efficient solution to a Nash bargaining game [103]. A Nash bargaining solution should satisfy certain axioms: Invariant to affine transformations or Invariant to equivalent utility representations, Pareto optimality, Independence of irrelevant alternatives, and Symmetry. Let us call u the utility function for player 1, v the utility function for player 2. Under these conditions, rational agents will choose what is known as the Nash bargaining solution. Namely, they will seek to maximize $|u(x) - u(d)||v(y) - v(d)|$, where $u(d)$ and $v(d)$ are the status quo utilities (i.e. the utility obtained if one decides not to bargain with the other player). The product of the two excess utilities is generally referred to as the Nash product. The Nash bargaining solution is the bargaining solution that maximizes the product of an agent's utilities on the bargaining set. The Nash bargaining solution, however, only deals with the simplest structure of bargaining. It is not dynamic (failing to deal with how Pareto outcomes are achieved). Instead, for situations where the structure of the bargaining game is important, a more mainstream game theoretic approach is useful. This can allow players' preferences over time and risk to be incorporated into the solution of bargaining games. It can also show how the details can matter. For example, the Nash bargaining solution for Prisoners' Dilemma is different from the Nash equilibrium.

Another typical bargaining solution is the Kalai-Smorodinsky bargaining solution. Independence of Irrelevant Alternatives can be substituted with an appropriate monotonicity condition, thus providing a different solution for the class of bargaining problems. This alternative solution was introduced by Ehud Kalai and Meir Smorodinsky [67]. It is the point that maintains the ratios of maximal gains. In other words, if player 1 could receive a maximum of g_1 with player 2's help (and vice-versa for g_2), then the Kalai-Smorodinsky bargaining solution would yield the point ϕ on the Pareto frontier such that $\phi_1 \ / \ \phi_2 = g_1 \ / \ g_2$.

In addition, bargaining mechanism design for non-cooperative negotiation appeared. The research in the economics community mainly focuses on computing agents' equilibrium strategies and the research in the AI part contributes to the development of software agents that negotiate on behalf of their users in realistic environments in which it is often impossible to compute agents' equilibrium strategies. The strategic bargaining has received more attention by Rubinstein's path-breaking work [121]. The study in the economics and AI community mainly focuses on computing agents' equilibrium strategies and contributes to the development of software agents that negotiate on behalf of their users in realistic environments in which it is often impossible to compute agents' equilibrium strategies.

Bargaining mechanism design generally focuses on bilateral monopoly, in which a buyer and a seller are bargaining over the price of an object (e.g., a good). The Myerson-Satterthwaite theorem [102] is one of the most remarkable negative results in economics. Informally, the theorem says that there is no efficient way for two parties to trade a good when they each have secret and probabilistically varying valuations for it, without the risk of forcing one party to trade at a loss. Myerson and Satterthwaite analyze bargaining as a static direct revelation game in which each player reports its type to a third party, and the third party chooses whether the object is transferred, and how much the buyer must pay.

Chatterjee and Samuelson [13] analyze a strategic game in which both players make offers simultaneously, and the trade occurs at a price between the two offers if the seller's offer is less than the buyer's offer. This game is closely related to the direct revelation game since it is static. Moreover, it can be shown that for a particular class of examples, the simultaneous-offers game implements the direct revelation game in which the outcome functions are chosen to maximize the players' utility.

Gupta and Livne's solution formally represents a reference point by replacing the conflict point as an outcome that both parties should attempt to improve jointly [48]. The solution proposed by Gupta and Livne is a point that lies on the Pareto-optimal line and connects this reference point with the maximum achievement of each party's aspiration levels.

The alternating-offers protocol was pioneered by Rubinstein [121] in a setting with complete information. The original alternating-offers protocol is designed for the simple discrete time bilateral single-issue negotiation and the allowed actions include offer and accept. The alternating-offers game represents a very general bargaining rule: at any time, a bargainer may make a new offer or accept the most recent offer of its opponent. The alternating-offers protocol captures the most important features of bargaining: bargaining consists of a sequence of offers and decisions to accept or reject these offers. The alternating-offers protocol has been widely used in the bargaining theory literature [122, 127].

Game theory is closely related to social choice theory [4]. Social choice theory specifies how the group should behave so that its actions are consistent with some postulate of rationality. In game theory, on the other hand, the rationality principle is imposed on the individual, not the group. Thus, social choice theory seeks to determine the expected group utility function like non-cooperative game theory, whereas game theory seeks first to determine the individual benefits for each alternative, before determining the group's benefit.

Arrow's impossibility theorem [4, 41, 119] is one of the relevant theorems for proving that no voting system can be designed that satisfies these three "fairness" criteria: "If every voter prefers alternative X over alternative Y, then the group prefers X over Y."; "If every voter's preference between X and Y remains unchanged, then the group's preference between X and Y will also remain unchanged (even if voters' preferences between other pairs like X and Z, Y and Z, or Z and W change)."; "There is no "dictator": no single voter possesses the power to always determine the group's preference."

There are several voting systems that side-step these requirements by using cardinal utility (which conveys more information than rank orders) and weakening the notion of independence (see the subsection discussing the cardinal utility approach to overcoming the negative conclusion). Arrow, like many economists, rejected cardinal utility as a meaningful tool for expressing social welfare, and so focused his theorem on preference rankings.

2.3.4 Heuristic Approaches

The major means of overcoming the aforementioned limitations of game theoretic models is to use heuristic methods. Such methods acknowledge that there is a cost associated with computation and decision making, and thus seek to search the negotiation space in a non-exhaustive fashion. This has the effect that heuristic methods aim to produce good, rather than optimal solutions.

The models are based on realistic assumptions; hence they provide a more suitable basis for automation and they can, therefore, be used in a wider variety of application domains; the designers of agents can use alternative, and less constrained, models of rationality to develop different agent architectures. For example, the heuristic approach is highly employed in the Automated Negotiation Agent Competition [61]. The central concern of this line of work is to model the agent's decision making heuristically during the course of the negotiation.

The space of possible agreements is quantitatively represented by contracts having different values for each issue. Each agent then rates these points in the space of possible outcomes according to some preference structure, captured by a utility function. Proposals are then offers over single points in this space of possible outcomes, and search terminates either when the time to reach an agreement has been exceeded or when a mutually acceptable solution, a point of intersection of the agents' acceptable outcomes sets, has been reached.

Whereas the protocol normatively describes the orderings of actions, the decision-making mechanisms describe the possible set of agent strategies in using the protocol. These strategies are captured by a negotiation architecture that is composed of responsive and deliberative decision mechanisms.

Faratin et al. [27] proposed a mechanism based on a linear combination of simple functions called tactics, which manipulate the utility of contracts. The mechanisms are subdivided into trade-off and issue manipulation mechanisms [28].

After that, Faratin et al. [29] presented a strategy called the trade-off strategy where multiple negotiation decision variables are traded-off against one another. The aim of this paper is to develop a heuristic computational model of the trade-off strategy and show that it can lead to an increased social welfare of the system. The algorithm itself operates by using the notion of fuzzy similarity to approximate the preference structure of the other negotiator and then uses a hill-climbing technique to explore the space of possible trade-offs for the one that is most likely to be acceptable.

Klein et al. [74] present the first negotiation protocols specifically for complex preference spaces. They focus on the nonlinear utility function, and describe a simulated annealing-based approach appropriate for negotiating such complex contracts that achieves near-optimal social welfare for negotiations with binary issue dependencies. This paper focuses on nonlinear utility functions, and introduces the heuristic search method to negotiation protocols.

Generally speaking, heuristic methods also have a number of characteristics.

- The models often select outcomes (deals) that are sub-optimal; this is because they adopt an approximate notion of rationality and because they do not examine the full space of possible outcomes.

- The models need extensive evaluation, typically through simulations and empirical analysis, since it is usually impossible to predict precisely how the system and the constituent agents will behave in a wide variety of circumstances.

2.3.5 Argumentation-based Approaches

The basic idea behind the argumentation-based approach is to allow additional information to be exchanged, over and above proposals. This information can be of a number of different forms, all of which are arguments that explain explicitly the opinion of the agent making the argument. Thus, in addition to rejecting a proposal, an agent can offer a critique of the proposal, explaining why it is unacceptable. This has the effect of identifying an entire area of the negotiation space as being not worth exploring by the other agent. Similarly, an agent can accompany a proposal with an argument that says why the other agent should accept it. This latter kind of argument makes it possible to change the other agent's region of acceptability (by altering its preferences), and also provides a means of changing the negotiation space itself. Without the ability to argue for the worth of a new element in the negotiation object, the receiving agent would not, in general, have any basis on which to determine its value [110].

The exact argumentation mechanism is mainly logic-based and builds on work in argumentation as an approach to handling defeasible reasoning. This makes it possible for agents to handle contradictory statements (which frequently occur during arguments) without collapsing into triviality, and allow conflicting arguments to be resolved. Using argumentation in real agents (as opposed to simple collections of logical statements) means handling the complexities of the agents' mental attitudes, communication between agents, and the integration of the argumentation mechanisms into a complex agent architecture.

The PERSUADER system was developed to model adversarial conflict resolution in the domain of labor relations, which can be multi-agent, multi-issue, and single or repeated negotiation encounters [137, 138, 139]. The system uses both case-based reasoning (CBR) and multi-attribute utility theory (MAUT) for conflict resolution problems. The model, with its iterative nature, is used to narrow the difference between the parties involved, takes into consideration changing environments, and models social reasoning (by modeling other parties' beliefs) as well as belief modification of parties. The PERSUADER system models both the iterative process of negotiation and the multi-issue nature of interactions. However, mediation is unsuitable for the problem domains of this research since negotiation is a mutual selection of outcomes. Furthermore, in the problem domains of this research, it is not necessary for the agents to have similar beliefs at the end of negotiation. Therefore, persuasion (operating over beliefs) is not a necessary condition for coordination in this problem domain.

However, two main areas of work remain in argumentation-based approaches. The first is in the definition of suitable argumentation protocols, that is, sets of rules that specify how agents generate and respond to arguments based upon what they know. As a result, we may end up with negotiators that are possibly rather inflexible in their argumentation stance. Since this seems rather limiting, we need to investigate this area more with the aim of discovering more flexible argumentation protocols than we currently have. The second main area of work is also related to argumentation protocols, and specifically the transition between the underlying negotiation protocol and the argumentation protocol. When is the right time to make this transition, when is it right to start an argument? Clearly, it only makes sense to engage in the complex business of argumentation when it will help the negotiation, but we need to translate this high-level notion of "rightness" into some more concrete decision criterion that can be built into our agents.

2.4 Contract Net Protocol (CNP)

The contract net protocol (CNP) [134] is a simple negotiation protocol for distributed problem-solving based on the notion of call for bids on markets. The original CNP protocol is for cooperative problem solving and it has a number of limitations. For example, in the original CNP model, a contractor can only respond to bids sequentially. However, in a multi-agent system, several managers may concurrently call for bids and it is important to give each contractor the opportunity to concurrently negotiate with multiple managers and optimize its utility. In addition, there is no counter-proposing in the CNP model.

It does not belong to the class of quantitative models of bargaining, although its operation closely resembles a market-like mechanism. The protocol focuses on the traditional problem of how to resolve disparate viewpoints in task allocation problems in a simulated distributed sensor network for acoustic interpretation. Nonetheless, it is included here because: i) it was traditionally the first negotiation protocol in DAI, ii) it models contracts and iii) its extension by Sandholm [126] brings it into the class of quantitative models of negotiation.

The original CNP protocol has been extended in different applications [1, 19, 73]. For instance, it has been applied to job dispatching among machines within a manufacturing plant [111], and to distributed meeting scheduling [130]. However, the protocol has a number of limitations borne out of the fact that it belongs to the CPS system. In particular, cooperation is an integral part of the protocol. There cannot be any conflict between the agents to start the CNP.

2.5 Auction Mechanism

Some auction mechanisms, especially combinatorial auctions, can enable large-scale collective decision making in nonlinear domains, and the studies of auction mechanism are highly efficient for automated negotiation protocols. As is well known, the auction mechanism has been widely used in resource allocation mechanisms in which agents bid for the best resources. However, the auction mechanism is for a very limited type when employing in automated multi-issue negotiations. Multi-attribute auctions, wherein buyers advertise their utility functions and sellers compete to offer the highest-utility bid, are also aimed at a fundamentally limited problem (a purchase negotiation with a single buyer) and require full revelation of preference information.

There are traditionally four types of auction that are used for single-item allocation. 1) In first-price auctions, bidders simultaneously submit their bids in a sealed envelope. The individual with the highest bid wins and pays its bidding price. 2) Second-price auctions (Vickrey auctions [143]) are similar to first-price sealed-bid auctions except that the winner pays a price equal to the exact amount of the second highest bid. 3) In English auctions, the price is steadily raised by the auctioneer with bidders dropping out once the price becomes too high. This continues until there remains only one bidder who wins the auction at the current price. 4) In Dutch auctions, the auctioneer begins with a high asking price that is lowered until some participant is willing to accept the price. The winning participant pays the last announced price. In addition, the VCG auction, which is a strategy-proof mechanism, is one of the most important auction mechanisms. The idea in VCG is that items are assigned to maximize the sum of utilities; then each player pays the "opportunity cost" that their presence introduces to all the other players [107]. However, strategy-proof mechanisms such as the well-known VCG auction are not necessarily strategy-proof and do not necessarily result in the most efficient usage in a dynamic situation [109] or Shill Bidders [154, 155].

A combinatorial auction is an auction in which participants can place bids on combinations of discrete items, or "packages," rather than just individual items or continuous quantities [18]. Combinatorial auctions have been widely studied by many researchers recently, but they present challenges compared to traditional auctions. In combinatorial auction schemes, a centralized controlling agent (the "auctioneer") assumes responsibility for determining which agents receive which resources based on the bids submitted by individual agents. However, the problem of deciding successful bids, i.e. the winner determination problem, is NP-hard [120], meaning that a polynomial-time algorithm to find the optimal allocation is unlikely ever to be found. In addition, the auctioneer may face significant computational overload due to a large number of bids with complex structures. Some researchers propose the novel efficient winner determination method in the combinatorial auctions using the heuristic algorithms [38, 158].

2.6 Distributed Constraint Optimization Problem

When we focus on the complex negotiations that are highly nonlinear, the area in which many authors have dealt with complexity characterization and measurement is the optimization. In fact, negotiation scenarios and optimization problems are often closely related, since there are many similarities in the ways both problem families are defined and addressed [129, 132, 149]. In particular, the Distributed Constraint Optimization Problem (DCOP) has many similarities to automated negotiation protocols with complexity because DCOP and automated negotiation assume the distributed decision makers in finding the solutions. On the other hand, even though automated negotiation seems to involve a straightforward constraint optimization problem, we have been unable to exploit existing work on high efficiency constraint optimizers. Such solvers attempt to find the solutions that maximize the weights of the satisfied constraints, but they do not account for the fact that agents are all self-interested.

Distributed constraint optimization is the distributed algorithm to constraint optimization. A constraint optimization problem can be defined as a regular constraint satisfaction problem in which constraints are weighted and the goal is to find a solution maximizing the weight of satisfied constraints. Alternatively, a constraint optimization problem can be defined as a regular constraint satisfaction problem augmented with a number of "local" cost functions. The aim of constraint optimization is to find a solution to the problem whose cost, evaluated as the sum of the cost functions, is maximized or minimized. The regular constraints are called hard constraints, while the cost functions are called soft constraints. These names illustrate that hard constraints are to be necessarily satisfied, while soft constraints only express a preference of some solutions (those having a high or low cost) over other ones (those having lower/higher cost) [152, 153]. A DCOP is a problem in which a group of agents must choose values for a set of variables such that the cost of a set of constraints over the variables is either minimized or maximized. Distributed Constraint Satisfaction is a framework for describing a problem in terms of constraints that are known and enforced by distinct agents.

The constraints are described on some variables with predefined domains, and have to be assigned to the same values by the different agents [133].

One remarkable work in DCOP is Adopt (Asynchronous Distributed OPTimization) [100]. Adopt is the first algorithm for DCOP that can find the optimal solution, or a solution within a user-specified distance from the optimal, using only localized asynchronous communication and polynomial space at each agent. Communication is local in that an agent does not send messages to every other agent, but only to neighboring agents. Another remarkable work in DCOP is the Distributed Pseudotree-Optimization Procedure (DPOP) [99, 112]. DPOP is based on dynamic programming. It is a utility propagation method, inspired by the sum-product algorithm, which is correct only for tree-shaped constraint networks. This algorithm requires a linear number of messages, whose maximal size depends on the induced width along the particular pseudotree chosen. Problems defined with this framework can be solved by any of the algorithms that are proposed for it. There has been some recent work on agent privacy in the DCOP field [44, 95].

2.7 Recent Literatures of Multi-issue Automated Negotiations

Lin et al. [86] explored a range of protocols based on mutation and selection on binary contracts. This paper does not describe what kind of utility function is used, nor does it present any experimental analyses, so it remains unclear whether this strategy enables sufficient exploration of utility space.

Barbuceanu and Lo [6] presented an approach based on constraint relaxation. However, this paper provides no experimental analysis and merely presents a small toy problem with 27 contracts.

Boutilier [9] presented a cooperative negotiation model for automated systems, through incremental utility elicitation. Using decentralized resource allocation as a problem setting, they emphasize the difficulty of eliciting complex utility functions and propose a strategy that requires only a small set of sampled utility function points in order to find near-optimal allocations. However, their model uses minimum regret, not Pareto-efficiency as an optimality criteria, and this paper doesn't handle multi-dimensional utility functions over multiple resources.

Debenham [20] proposed a multi-issue bargaining strategy that models iterative information gathering that takes place during negotiation. However, these models are not explicitly designed to address the problem of complex and high dimensional negotiations.

Lai et al. [78] presented a protocol for multi-issue problems for bilateral negotiations. This paper presents a decentralized model that allows self-interested agents to reach "win-win" agreements in a multi-attribute negotiation. The model is based on an alternating-offer protocol. Experimental analysis shows agents can reach near Pareto optimal agreements in quite general situations following the model where agents may have complex preferences on the attributes and incomplete information. Actually, this model does not require the presence of a mediator.

Robu et al. [65, 117, 118] presented the utility graph for issue interdependencies of binary-valued issues. Utility graphs are inspired by graph theory and probabilistic influence networks to derive efficient heuristics for non-mediated bilateral negotiations about multiple issues. The idea is to decompose highly non-linear utility functions in sub-utilities of clusters of inter-related items. They show how utility graphs can be used to model an opponent's preferences. In this approach agents need prior information about the maximal structure of the utility space to be explored. However, our approach has the advantage that outcomes can be reached without any prior information and that it is not restricted to binary-valued issues.

Bosse et al. [8] illustrated on experiments in (human) multi-issue negotiation and their analysis, and presents a generic software environment supporting such an analysis. The agents conduct one-to-one negotiations, in which the values across multiple issues are negotiated on simultaneously. To analyze such negotiation processes, the user can enter any formal property deemed useful into the system and use it to automatically check this property in given negotiation traces. This paper presents the results of applying this system in the analysis of empirical traces obtained from an experiment in multi-issue negotiation about second hand cars. In the experiment the efforts of 74 humans negotiating against each other have been analyzed.

Gerding et al. [42] proposed a negotiation mechanism where the bargaining strategy is decomposed into a concession strategy and a Pareto-search strategy. However, these papers also focus on bilateral multi-issue negotiations.

Jonker et al. [65] proposed a negotiation model called ABMP that can be characterized as cooperative one-to-one multi-criteria negotiation in which the privacy of both parties is protected as much as desired. Hindriks et al.[53] proposed an approach based on a weighted approximation technique to simplify the utility space. The resulting approximated utility function without dependencies can be handled by negotiation algorithms that can efficiently deal with independent multiple issues, and has a polynomial time complexity [66].

Hindriks et al. [54] proposed a checking procedure to mitigate this risk and showed that by tuning this procedure's parameters, outcome deviation can be controlled. These studies reflect interesting viewpoints, but they only focused on bilateral trading or negotiations. These approaches is efficient, however, the utility function of this approaches is different from the one based on multi-dimensional constraints.

Fatima et al. [31, 32] presented an algorithm to produce equilibrium strategies in multi-issue bargaining with uncertain reserve prices. By exploiting backward induction, their algorithm searches agents' strategy space from the deadline to the beginning of negotiation with the initial beliefs. Once the optimal strategies at the beginning of negotiation have been found, the system of beliefs are designed to be consistent with them. However, the optimization in their approach is myopic since it did not take into account its information effects. As a result, the strategies found by their approach are not guaranteed to be sequentially rational given the designed system of beliefs[40]. These papers focused on bilateral multi-issue negotiations.

Yager [151] a mediated negotiation framework for multi-agent negotiation was presented. This framework involves a mediation step in which the individual preference functions are aggregated to obtain a group preference function. The main interest is focused on the implementation of the mediation rule where they allow a linguistic description of the rule using fuzzy logic. A notable feature of their approach is the inclusion of a mechanism rewarding the agents for being open to alternatives other than simply their most preferred. The negotiation space and utility values are assumed to be arbitrary (i.e. preferences can be uncorrelated). However, the set of possible solutions is defined a priori and is fixed. Moreover, the preference function needs to be provided to the mediation step in the negotiation process, and pareto-optimality is not considered. Instead, the stopping rule is considered, which determines when the rounds of mediation stop.

Li et al. [81] proposed a method in which the mediator searches for a compromise direction based on an Equal Directional Derivative approach and computes a new tentative agreement in bilateral multi-issue negotiations. However, this method only focused on multilateral negotiation.

Zhang [156] presents an axiomatic analysis of negotiation problems within task-oriented domains (TOD). In this paper, three classical bargaining solutions (Nash solution, Egalitarian solution, Kalai-Smorodinsky solution) coincide when they are applied to a TOD with mixed deals but diverge if their outcomes are restricted to pure deals.

An et al. [3] proposed the design and implementation of a negotiation mechanism for dynamic resource allocation problem in cloud computing. Multiple buyers and sellers are allowed to negotiate with each other concurrently and an agent is allowed to decommitment from an agreement at the cost of paying a penalty.

Lin et al. [82, 83] focus on the Expert Designed Negotiators (EDN) which is the negotiations between humans and automated agents in real-life. In addition, the tools for evaluating automatic agents that negotiate with people were proposed. These studies include some efficient results from extensive experiments involving many human subjects and PDAs.

KEMNAD [94] develops a new methodology for building a negotiating agent by assembling these components rather than reinventing the wheel each time. Moreover, since these patterns are identified from a wide variety of existing negotiating agents, they can also improve the quality of the final systems developed.

2.8 Competition for Negotiating Agents

Motivated by the challenges of bilateral negotiations between people and automated agents, we organized the Automated Negotiating Agents Competition (ANAC) [61]. The purpose of the competition is to facilitate research in the area of bilateral multi-issue closed negotiation. The competition was based on the Genius environment [52, 84], which is a **G**eneral **E**nvironment for **N**egotiation with **I**ntelligent multi-purpose **U**sage **S**imulation. It allows easy development and integration of existing negotiating agents, and can be used to simulate individual negotiation sessions, as well as tournaments between negotiating agents in various negotiation scenarios. In this competition, we only consider bilateral negotiations, i.e. negotiation between two parties. The domain is common knowledge to the negotiating parties and stays fixed during a single negotiation session. The interaction between negotiating parties is regulated by a negotiation protocol that defines the rules of how and when proposals can be exchanged. We use the alternating-offers protocol for bilateral negotiation [121], in which the negotiating parties exchange offers in turns.

The Trading Agent Competition (TAC) is also related to automated agent negotiation. For 2010, TAC is divided into three games: TAC SCM (TAC Supply Chain Management), TAC/AA (TAC Ad Auction), and CAT (TAC Market Design) [45, 108, 136, 147]. TAC SCM was designed to simulate a dynamic supply chain environment. Agents have to compete to secure customer orders and components required for production. In the TAC/AA, game entrants design and implement bidding strategies for advertisers in a simulated sponsoring environment. The agents have to bid against each other to get an ad placement that is related to certain keyword combinations in a web search tool. The CAT Competition is a reverse of the normal TAC game: as an entrant you define the rules for matching buyers and sellers, while the trading agents are created by the organizers of the competition.

The Agent Reputation Trust (ART) Competition [22, 39] is also a negotiating agent competition with a testbed that allows the comparison of different strategies. The ART competition simulates a business environment for software agents that use the reputation concept to buy advice about paintings. Each agent in the game is a service provider responsible for selling its opinions when requested. The agent can exchange information with other agents to improve the quality of their appraisals. The challenge is to perceive when an agent can be trusted and to establish a trustworthy reputation.

2.9 Conclusion

This chapter showed the extensive related works of this thesis including Automated Negotiations, Bargaining theory, Auction Mechanism, Optimization Techniques, and recent studies of multi-issue automated negotiations. In particular, this chapter focused on studies of multi-agent systems.

We first described the summary of multi-agent systems. Then we discussed the related works on automated negotiation protocols. We also described the properties of designing negotiation protocols and classification of negotiation protocols. In addition, automated negotiation protocols based on game theoretic approaches, heuristic approaches, and Argumentation-based approaches were compared. Contract Net Protocol (CNP), Auction Protocol, and Distributed Constraint Optimization Problem (DCOP) were especially focused on. We compared our work in this thesis with the state of the art.

3

Negotiation with Highly Nonlinear Utility Functions

3.1 Introduction

In this chapter, we describe a model of nonlinear multi-issue negotiation, and a bidding-based negotiation protocol (Basic Bidding) designed for multiple-issue negotiation protocol suited for agents with highly nonlinear utility functions. In this chapter, constraint-based utility function is focused because this expression produces a "bumpy" and highly nonlinear utility function. In Basic Bidding Algorithm, agents generate bids by sampling their own utility functions to find local optima, and then using constraint based bids to compactly describe regions that have large utility values for that agent. These techniques make bid generation computationally tractable even in large utility spaces. A mediator then finds a combination of bids that maximizes social welfare. This chapter is mainly based on the Ito et al. [56]

The remainder of the chapter is organized as follows. In 3.2, we describe a model of non-linear multi-issue negotiation. In 3.3, we describe a bidding-based negotiation protocol designed for such contexts. Finally, we conclude this chapter.

3.2 Multi-issue Negotiation Environment

3.2.1 Basic Model of Multi-issue Negotiation

Definition 1: Agents and Mediator. N agents $(a_1, \ldots, a_N)$ want to reach an agreement with a mediator who manages the negotiation from a man-in-the-middle position.

Definition 2: Issues under negotiation. There are M issues $(i_1, \ldots, i_M)$ to be negotiated.[1][2]

Definition 3: Contract Space. The negotiation solution space is defined by the values that the different values may take. To simplify, we assume that the issue takes a value drawn from the domain of integers $[0, X]$:

$$D = [0, X]^M$$

Definition 4: Contract or potential solution.

$$\vec{s} = (s_1, ..., s_M)$$

A contract is represented by a vector of issue values. Each issue s_j has a value drawn from the domain of integers $[0, X]$ $(1 \leq j \leq M)$.(*i.e.* $s_j \in \{0, 1, , \ldots, X\}$)[3].

3.2.2 Constraint-based Complex Utility Model

In this thesis, some protocols and experiments rely on the constraint-based utility model. In other words, an agent's utility function, in our formulation, is described in terms of constraints. This expression produces a "bumpy" nonlinear utility function and a crucial departure from previous efforts on multi-issue

[1] The number of issues represents the number of dimensions in the utility space.

[2] The issues are shared: all agents are potentially interested in the values for all M issues.

[3] A discrete domain can come arbitrarily close to a 'real' domain by increasing its size. As a practical matter, many real-world issues that are theoretically 'real' numbers (delivery date, cost) are discretized during negotiations.

negotiation, where contract utility is calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyper planes with a single optimum. Therefore, this utility model is a key point of assuming negotiations with highly nonlinear utility functions.

Definition 5: Constraint.

$$c_k \in C \ (1 \leq k \leq l).$$

There are l constraints in an agent's utility space. Each constraint represents a region in the contract space with one or more dimensions and an associated utility value.

Definition 5-1: Constraint Value. Constraint c_k has value $w_a(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}$.

Definition 5-2: Constraint Region. Function $\delta_a(c_k, i_j)$ is a region of i_j in c_k. $\delta_a(c_k, i_j)$ is $\emptyset$ if c_k has no region regarded as i_j.

Definition 5-3: The Number of Terms in the Constraint. Function $\epsilon_a(c_k)$ is the number of terms in c_k.

Definition 6: Utility function.

$$u_a(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_a(c_k, \vec{s}),$$

where $x(c_k)$ is a set of possible contracts (solutions) of c_k.

An agent's utility for contract $\vec{s}$ is defined as the sum of the utility for all the constraints it satisfies.

Definition 7: The relationship between agents and constraints Every agent has *its own, typically unique*, set of constraints.

This expression produces a "bumpy" nonlinear utility function with high points where many constraints are satisfied and lower regions where few or no constraints are satisfied. This represents a crucial departure from previous efforts on

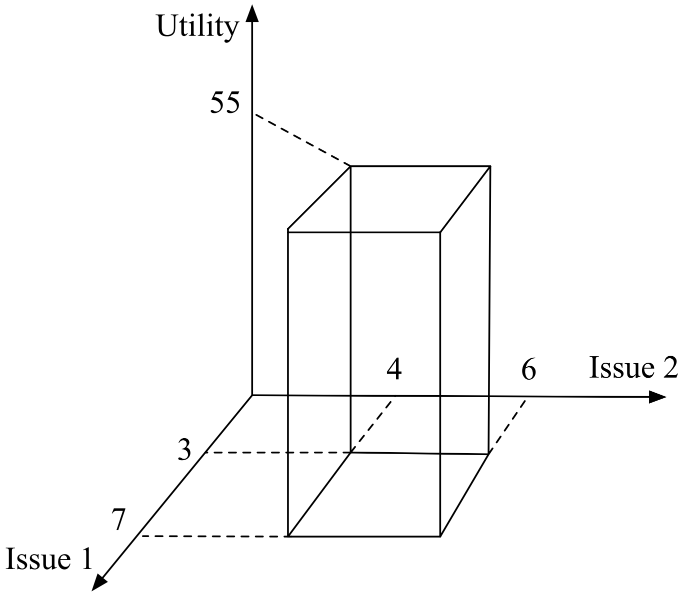

Figure 3.1: Example of constraint

multi-issue negotiation, where contract utility is calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyper planes with a single optimum.

Figure 3.2 shows an example of a utility space generated via a collection of binary constraints involving Issues 1 and 2. In addition, the number of terms is two in Figure 3.2. Figure 3.1, for example, which has a value of 55, holds if the value for Issue 1 is in the range $[3, 7]$ and the value for Issue 2 is in the range $[4, 6]$. The utility function is highly nonlinear with many hills and valleys. For our work, we assume that many real-world utility functions are more complex than this, involving more than two issues as well as higher-order (e.g. trinary and quaternary) constraints.

This constraint-based utility function representation allows us to capture the issue interdependencies common in real world negotiations. The constraint in Figure 3.2, for example, captures the fact that a value of 4 is desirable for issue 1 if issue 2 has the value 4, 5 or 6. Note, however, that this representation is also capable of capturing linear utility functions as a special case (they can be

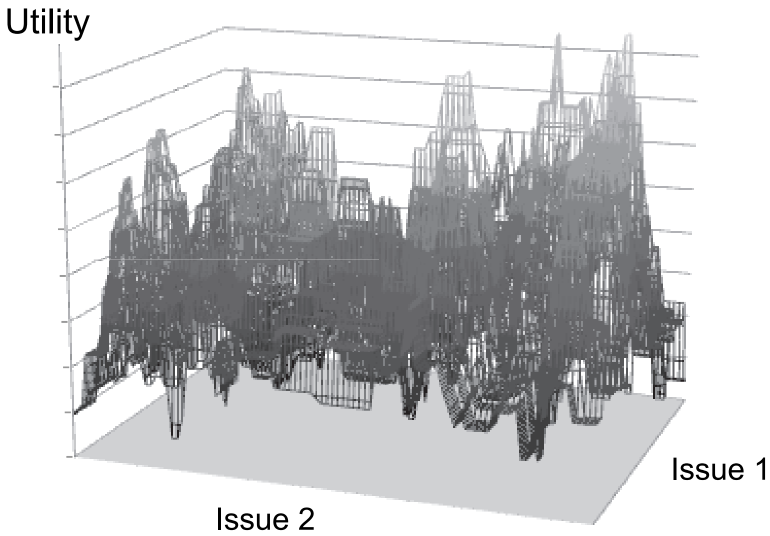

Figure 3.2: An example of a nonlinear utility space

captured as a series of unary constraints). A negotiation protocol for complex contracts can, therefore, handle linear contract negotiations.

We assume, as is common in negotiation contexts, which agents do not share their utility functions with each other, in order to preserve a competitive edge. It will generally be the case, in fact, that agents do not fully know their desirable contracts in advance, because each own utility functions are simply too large. If we have 10 issues with 10 possible values per issue, for example, this produces a space of 10^{10} (10 billion) possible contracts, too many to evaluate exhaustively. Agents must thus operate in a highly uncertain environment.

3.2.3 Objective Function

The objective function for our protocol can mainly be described as follows:

$$\arg\max_{\vec{s}} \sum_{a \in N} u_a(\vec{s}).$$

Our protocol, in other words, tries to find contracts that maximize social welfare, *i.e.*, the total utilities for all agents. Such contracts, by definition, will also be Pareto-optimal.

It is of course theoretically possible to gather all the individual agents' utility functions into one central place and then find all optimal contracts using such well-known nonlinear optimization techniques as simulated annealing or evolutionary algorithms. However, we do not employ such centralized methods for negotiation purposes because we assume, as is common in negotiation contexts, that agents prefer not to share their utility functions with each other, in order to preserve a competitive edge.

3.3 Basic Bidding Protocol

In a previous work [56], agents reach an agreement based on the following steps. It is called a **basic bidding protocol**. Basic Bidding protocol is one of the remarkable result focusing on the complex automated negotiation with highly nonlinearity. In fact, the proposed automated negotiation protocols in this thesis are compared with this basic bidding protocol for evaluation.

The basic bidding protocol consists of the following four steps:

[**Step 1: Sampling**] Each agent samples its utility space in order to find high-utility contract regions. A fixed number of samples are taken from a range of random points, drawing from a uniform distribution. Note that, if the number of samples is too low, the agent may miss some high utility regions in its contract space, and thereby potentially end up with a sub-optimal contract.

[**Step 2: Adjusting**] There is no guarantee, of course, that a given sample will lie on a locally optimal contract. Each agent, therefore, uses a nonlinear optimizer based on simulated annealing to try to find the local optimum in its neighborhood. Figure 3.3 exemplifies this concept. In this figure, a black dot is a sampling point and a white dot is a locally optimal contract point.

[**Step 3: Bidding**] For each contract $\vec{s}$ found by adjusted sampling, an agent evaluates its utility by summation of values of satisfied constraints. If that utility is larger than the reservation value δ, then the agent defines a bid that covers all

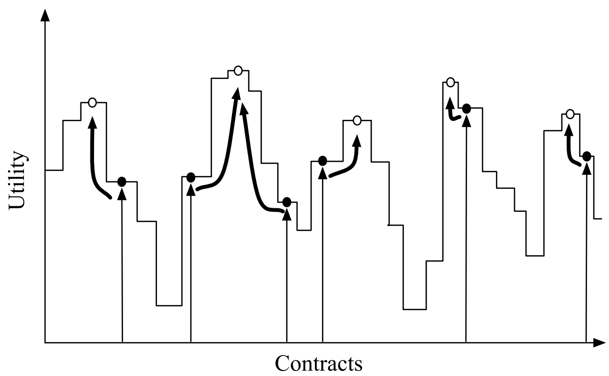

Figure 3.3: Adjusting the Sampled Contract Points

the contracts in the region that has that utility value. This is easy to do: the agent need merely find the intersection of all the constraints satisfied by that $\vec{s}$.
[**Step 4: Deal identification**] The mediator identifies the final contract by finding all the combinations of bids, one from each agent, that are mutually consistent, *i.e.*, that specify overlapping contract regions[1]. If there is more than one such overlap, the mediator selects the one with the highest summed bid value (and thus, assuming truthful bidding, the highest social welfare) (see Figure 3.4). Each bidder pays the value of its winning bid to the mediator.

The mediator employs breadth-first search with branch cutting to find social-welfare-maximizing overlaps.

It is easy to show that, in theory, this approach can be guaranteed to find optimal contracts. If every agent exhaustively samples every contract in its utility space, and has a reservation value of zero, then it will generate bids that represent the agent's complete utility function. The mediator, with the complete utility functions for all agents in hand, can use exhaustive search over all bid combinations to find the social welfare maximizing negotiation outcome. But this approach is only practical for very small contract spaces. The computational

[1]A bid has an acceptable region. For example, if a bid has a region, such as [0,2] for issue1, [3,5] for issue2, the bid is accepted by a contract point [1,4], which means issue1 takes 1, issue2 takes 4. If a combination of bids, i.e. a solution, is consistent, there are definitely overlapping region. For instance, a bid with regions (Issue1,Issue2) = ([0,2],[3,5]), and another bid with ([0,1],[2,4]) is consistent.

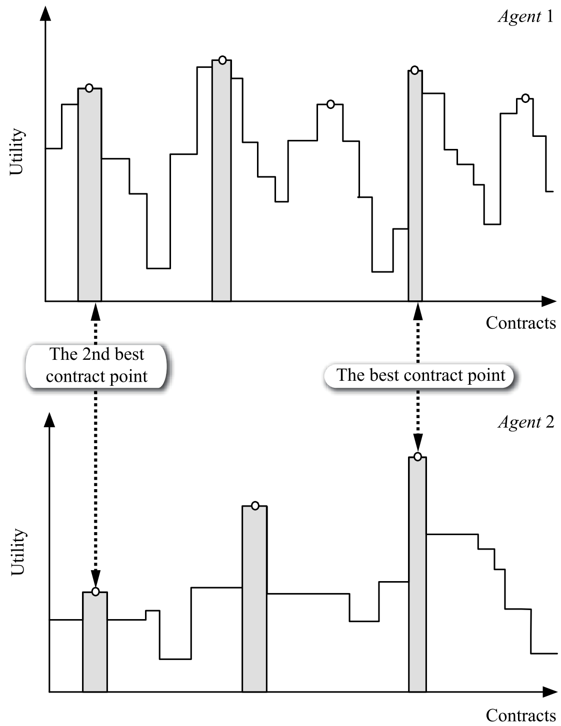

Figure 3.4: Deal Identification

cost of generating bids and finding winning combinations grows rapidly as the size of the contract space increases. As a practical matter, we introduce the threshold to limit the number of bids the agents can generate. Thus, deal identification can terminate in a reasonable amount of time.

3.4 Conclusion

In this chapter, we have presented a model of nonlinear multi-issue negotiation, constraint based utility space, and a basic bidding protocol designed for the important challenge of negotiation with highly nonlinear utility functions. In the basic bidding protocol, agents generate bids by sampling their own utility functions to find local optima, and then using constraint based bids to compactly describe regions that have large utility values for that agent.

4

Threshold Adjusting Mechanism

4.1 Introduction

The existing works about the automated negotiation protocols with nonlinear utility function have not yet concerned about agents' private information. Such private information should be kept as much as possible in their negotiation. In this chapter, we propose a threshold adjusting mechanism. First agents make bids that produce more utility than the common threshold value based on the basic bidding protocol [56]. Then the mediator asks each agent to reduce its threshold based on how much each agent opens its private information to the others. Each agent makes bids again above the threshold. This process continues iteratively until agreement is reached or no solution. Our experimental results show that our method substantially outperforms the existing negotiation methods on the point of how much agents have to open their own utility space.

The remainder of this chapter is organized as follows. In 4.2, we propose a threshold adjusting mechanism that helps agents to keep their private information secret as much as possible. In 4.3, we present experimental assessment of this protocol. Finally, we conclude with a discussion.

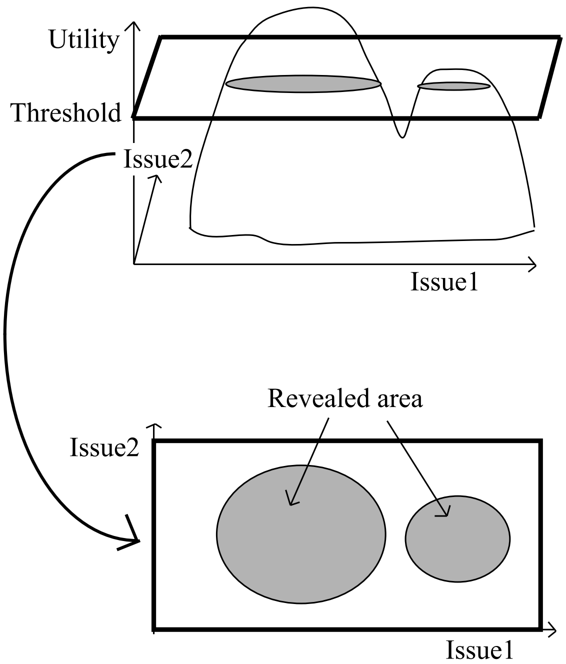

Figure 4.1: Revealed Area

4.2 Threshold Adjusting Mechanism

4.2.1 The Outline of the Threshold Adjusting Mechanism

The main idea of the threshold adjusting mechanism is that if an agent reveals the larger area of his utility space, then he can persuade the other agents. On the other hand, an agent who reveals the small area of his utility space, he should adjust his threshold to agree with if no agreement is achieved. The revealed area can be defined how the agent reveals his utility space on his threshold value. The threshold value is defined at the same value beforehand. Then the threshold values are changed by each agent based on the amount of the revealed area afterwards. Figure 4.1 shows the concept of the revealed area of agent's utility space. If the agent decreases the threshold value, then this means that he reveals his utility space more.

Figure 4.2 shows an example of the threshold adjusting process among 3 agents. The upper figure shows the thresholds and the revealed areas before

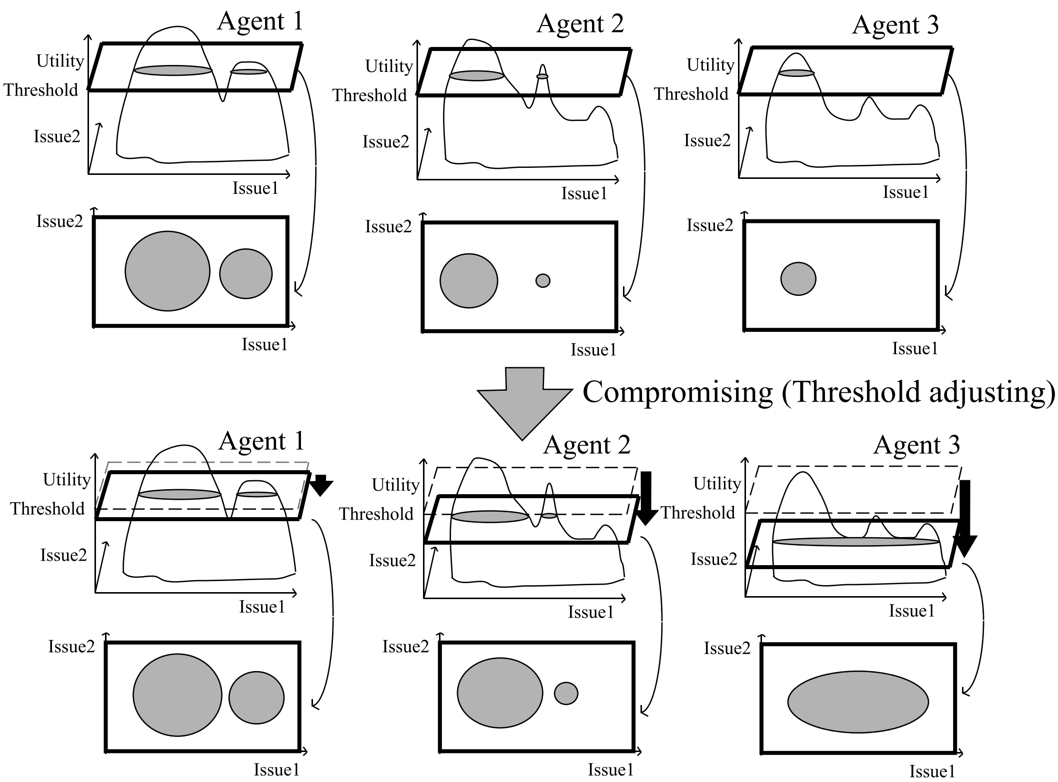

Figure 4.2: The Threshold Adjusting Process

adjusting the threshold. The bottom figure shows the thresholds and the revealed areas after adjusting the threshold. In particular, in this case, agent 3 revealed the small amount of his utility space. The amount of agent 3's revealing utility space in this threshold adjusting is largest among these 3 agents. In the protocol, this process is repeated until an agreement is achieved or until they could not find any agreement. The exact rate of the amount of revealed utility space and the amount of decreasing the threshold is defined by the mediator or the mechanism designer.

In the former paper [56], we did not define any external loop of these steps. Threshold Adjustment protocol is the first that proposed the external loop for an effective consenting mechanism. The details of the threshold adjusting mechanism is shown as follows:

Algorithm 1 threshold_adjustment( )

- Ar: Area Range of each agent ($Ar = \{Ar_0, Ar_1, ..., Ar_n\}$)
- *bid_generation_with_SA*(Th_i,V,SN,T,B_i): An Agent samples, adjusts and bids based on the basic bidding protocol.
- *search_solution*(B): The mediator employs breadth-first search with branch cutting to find social-welfare-maximizing overlaps. This step is based on the winner determination step of basic bidding protocol.

loop
    $i := 1$, $B := \emptyset$
    while $i < |Ag|$ **do**
        bid_generation_with_SA(Th_i,V,SN,T,B_i)
    end while
    $SC := \emptyset$
    $maxSolution$:= search_solution(B)
    if find $maxSolution$ **then**
        $maxSolution$:= getMaxSolution(SC)
        break loop
    else if all agent can lower the threshold **then**
        $i := 1$
        $SumAr := \Sigma_{i \in |Ag|}\ Ar_i$
        while $i < |Ag|$ **do**
            $Th_i := Th_i - C * (SumAr - Ar_i)/SumAr$
            $i := i + 1$
        end while
    else
        break loop
    end if
end loop
return $maxSolution$

4.2.2 Incremental Deal Identification

The threshold adjusting process shown in the previous section could reduce the computational cost of deal identification in step 4 of Basic Bidding Protocol. The original step 4 of Basic Bidding Protocol requires an exponential computational cost because the computation is actually combinatorial optimization. In the new threshold adjusting process, agents incrementally reveal their utility spaces as bids. Thus, for each round, the mediator only computes the new combinations of bids that submitted newly in that round. This process actually reduces the computational cost.

4.3 Experimental Result

4.3.1 Setting

We conducted several experiments to evaluate the effectiveness of our approach. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions. We compare our new threshold adjusting protocol and the existing protocol without adjusting the threshold in terms of optimality and privacy.

In the experiments on optimality, for each run, we applied an optimizer to the sum of all the agents' utility functions to find the contract with the highest possible social welfare. This value was used to assess the efficiency (*i.e.*, how closely optimal social welfare was approached) of the negotiation protocols. To find the optimum contract, we used simulated annealing (SA) because exhaustive search became intractable as the number of issues grew too large. The SA initial temperature was 50.0 and decreased linearly to 0 over the course of 2500 iterations. The initial contract for each SA run was randomly selected.

In terms of privacy, the measure is the range of revealed area. Namely, if an agent reveals one point of the gird of utility space, this means he lost 1 privacy unit. If he reveals 1000 points, the he lost 1000 privacy.

The parameters for our experiments were as follows: Number of agents is $N = 3$. Number of issues is 2 to 10. Domain for issue values is $[0, 9]$. Utility function for per agent has 10 unary constraints, 5 binary constraints, 5 trinary constraints, etc.

(a unary constraint relates to one issue, an binary constraint relates to two issues, and so on). The maximum value for a constraint is $100 \times (Number\ of\ Issues)$. Constraints that satisfy many issues thus have, on average, larger weights. This seems reasonable for many domains. In meeting scheduling, for example, higher order constraints concern more people than lower order constraints, so they are more important for that reason. The maximum width for a constraint is 7. The following constraints, therefore, would all be valid: issue $1 = [2, 6]$, issue $3 = [2, 9]$ and issue $7 = [1, 3]$.

We compared three types of protocols.

No Threshold Adjustment: The Basic Bidding Protocol is applied [56]. This protocol exhaustively explores the whole utility space.

No Threshold Adjustment with limitation: The Basic Bidding Protocol with bids limitations is applied [56]. This protocol exhaustively explores the whole utility space. However, the number of agent's bids is limited to $\sqrt[N]{6400000}$.

Threshold Adjustment: Our proposed adjusting protocol. This protocol does not have the explicit limitation of the number of bids. "Threshold adjustment (50)", "Threshold adjustment (200)", and "Threshold adjustment (400)" mean mechanisms with the threshold adjustment. Each mechanism determines the decreasing amount of the threshold by $50 \times (SumAr - Ar_i)/SumAr$, $200 \times (SumAr - Ar_i)/SumAr$, and $400 \times (SumAr - Ar_i)/SumAr$, respectively. $SumAr$ means the sum of all agents' revealed areas. Ar_i means $agent_i$'s revealed area.

The number of samples taken during random sampling is $(Number\ of\ Issues) \times 200$. Annealing schedule for sample adjustment: initial temperature 30, 30 iterations. Note that it is important that the annealer not run too long or too 'hot', because then each sample will tend to find the global optimum instead of the peak of the optimum nearest the sampling point.

The threshold agents used to select which bids to make in starts with 900 and decreases until 200 in the threshold adjusting mechanism. The protocol without the threshold adjusting process defines the threshold as 200. The threshold is used to cut out contract points that have low utility.

The limitation on the number of bids per agent: $\sqrt[N]{6400000}$ for N agents. It was only practical to run the deal identification algorithm if it explored no more than about 6400,000 bid combinations, which implies a limit of $\sqrt[N]{6400000}$ bids per agent, for N agents.

In our experiments, we ran 100 negotiations in every condition. Our code was implemented in Java 2 (1.5) and run on a core 2 duo processor iMac with 1.0GB memory under Mac OS X 10.4.

4.3.2 Experimental Results

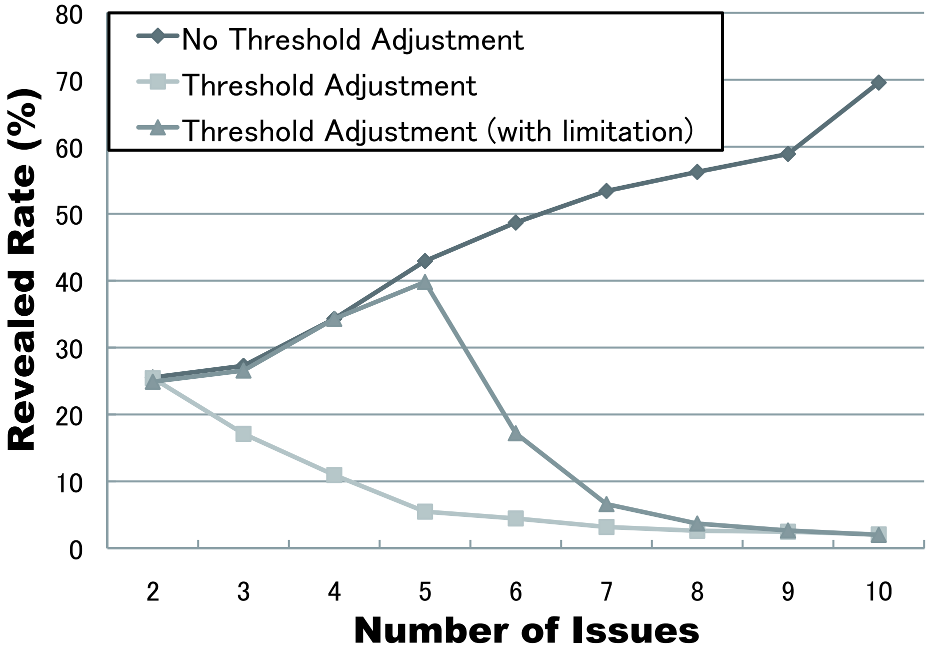

Figure 4.3: Revealed Rate

Figure 4.3 shows the optimality of the comparable mechanisms, one with the threshold adjustment (with bid limitation), one without the threshold adjustment and bid limitation, and one without the threshold adjustment with bid limitation. The revealed rate is defined by *(Revealed rate) = (Revealed area) / (Whole area of utility space)*.

The mechanism without both of the threshold adjustment and bid limitation (No Threshold Adjustment) increases the revealed rate. This means that if we do not use the threshold adjustment and bid limitation, then agents need to reveal their utility space much more than the other mechanisms. We found that bid limitation can show the nice effects to keep the increasing amount of revealed rate small. The mechanism with bid limitation but without the threshold adjustment shown by triangles starts decreasing when the number of issue is 5, namely bid limitation starts being active. Compared with the above two mechanisms, the mechanism with the threshold adjustment proposed in this chapter drastically decreases the amount of the revealed rate.

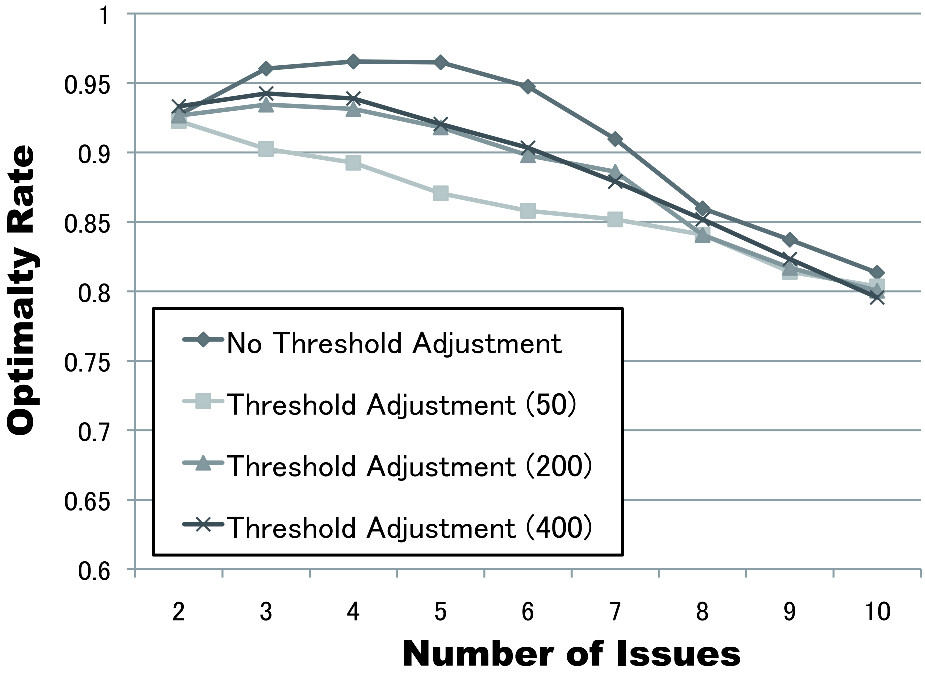

Figure 4.4: Optimality

As we show in the previous paragraph, our proposed threshold adjustment mechanism can effectively reduce the revealed rates. We then show the optimality of our proposed mechanism is quite competitive compared with the other mechanisms in Figure 4.4. As we can see in Figure 4.4, in terms of the optimality, the difference between "No Threshold Adjustment" and "Threshold Adjustments" is small. At most the difference is around 0.1 around 3 issues to 7 issues. When the threshold decreasing amount is not large, say 50, agents could miss the agreement points that have larger total utilities. This occurs when some agents have higher utility on the agreement point but other agents have very lower utility on the agreement point. “No threshold adjustment” mechanism makes agents to submit all agreement points that have larger utility than the minimum threshold. Thus, “No threshold adjustment” can find such cases. But “Threshold adjustment” mechanisms fail to capture such cases when the decreasing amount is smaller.

Figure 4.5(1) shows the number of bids for each mechanism. The number of bids means the utility space needed to be explored and the time needed to find the possible deal. The number of bids of “No Threshold Adjustment” increases exponentially. Actually, our program fails to compute the combinations completely at more than 6 issues when using “No Threshold Adjustment”. Figure 4.5(1) just shows the ideal numbers for “No Threshold Adjustment”. Our protocol, “Threshold Adjustment”, drastically reduces the number of bids.

Figure 4.5(2) shows the same results as Figure 4.5(1) without “No Threshold Adjustment”. “No Threshold Adjustment with limitation” manually limits the number of bids. The increase of the number of bids stops at the limitation defined above. On the other hand, as shown in the Figure 4.5(2), “Threshold Adjustment” succeeded to reduce the number of bids drastically.

Fairness on revealed areas is defined as the deviation of the amount of revealed areas for each agent. Thus, to confirm the fairness on revealed areas in our mechanism, we measured average standard deviations on agents’ revealed areas. Figure 4.6 shows the average standard deviations in “No Threshold Adjustment with limitation” and “Threshold Adjustment”. Here without loss of generality we assume the number of issues is 3. Comparing “Threshold Adjustment” with “No threshold Adjustment” the average standard deviation of the threshold adjustment is much lower than that of the protocol without threshold adjustment. Thus, the threshold adjustment could achieve fair results on the amount of revealed area. Also, the standard deviation increases as the number of agents increases and there are many kinds of agents.

Figure 4.7 compares the required rounds and the revealed rates for different decreasing amounts, 50 and 200. The left graph of Figure 4.7 demonstrates that the decreasing amount is small, 50, then the number of rounds could be larger.

On the other hand, in the right graph of Figure 4.7, if the decreasing amount is small, then the revealed rate is relatively small.

4.4 Conclusion

In this chapter, we proposed a threshold adjusting mechanism in very complex negotiations among software agents. In very complex negotiations, we assume agents have to do interdependent multi-issue negotiation. The threshold adjusting mechanism can facilitate agents to reach an agreement while keeping their private information as much as possible. The experimental results demonstrated that the threshold adjusting mechanism can reduce the amount of private information that is required for an agreement among agents, and reduce the computational complexity of the winner determination part.

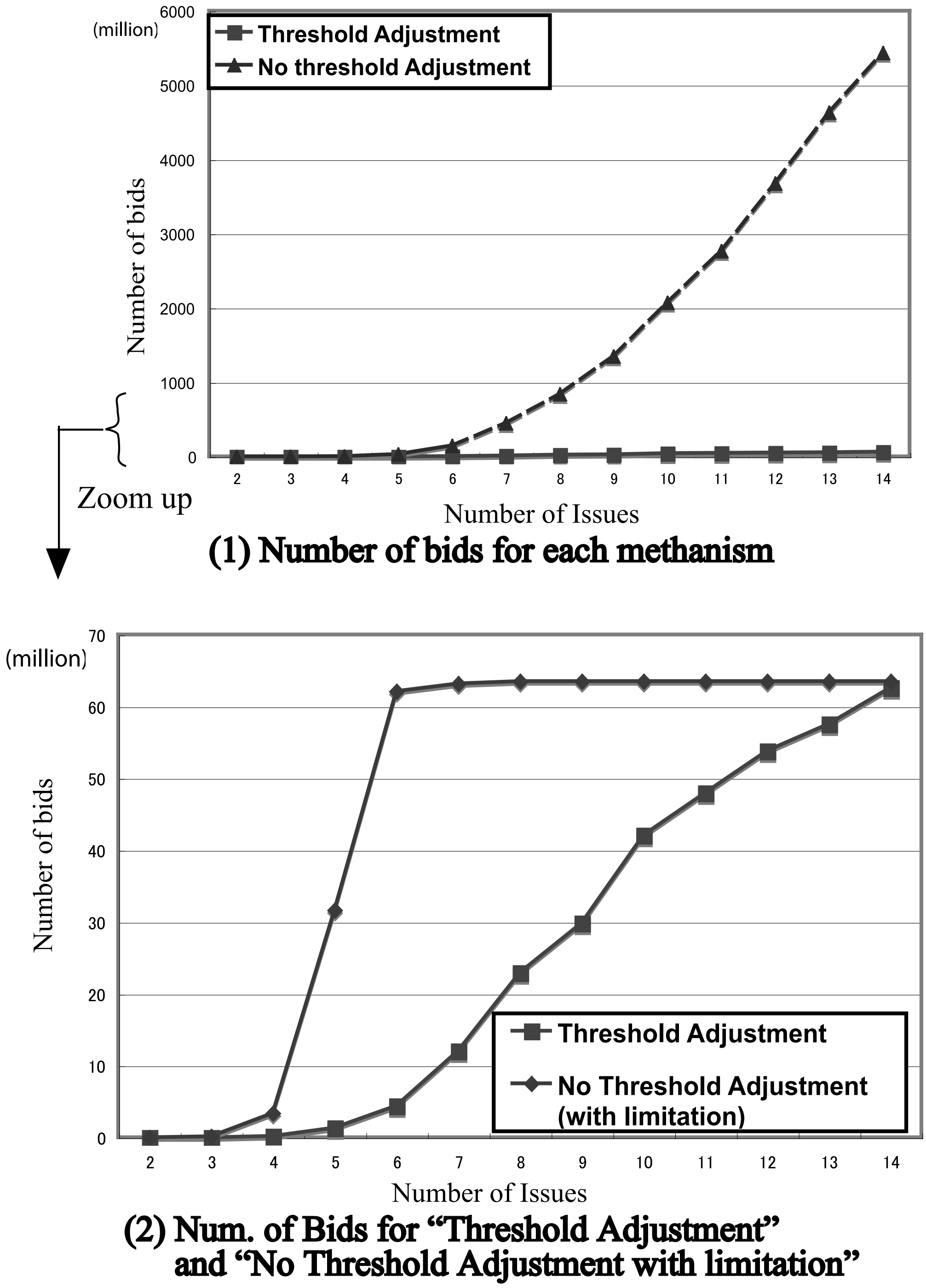

Figure 4.5: The number of bids

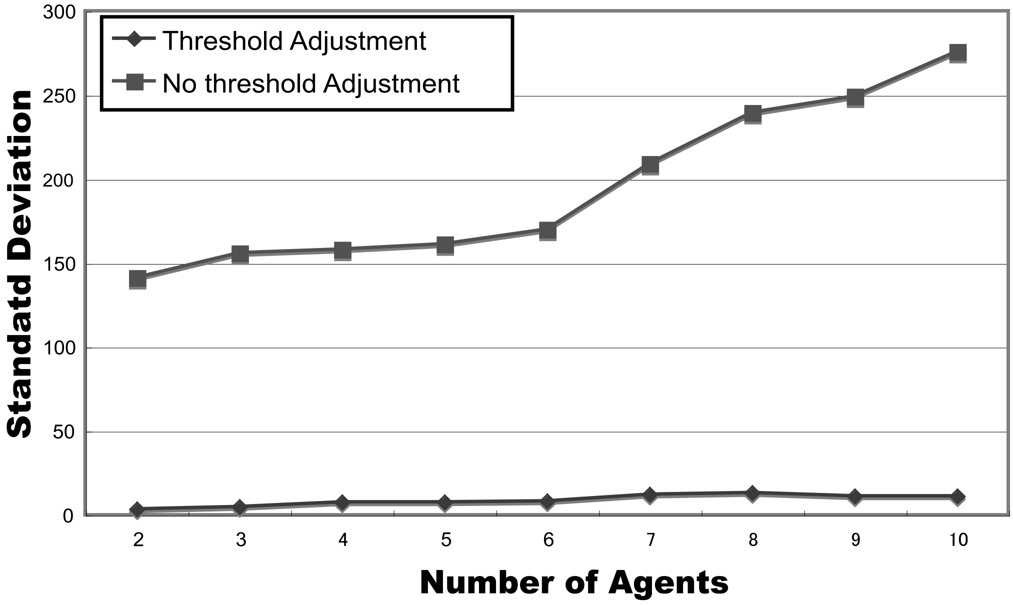

Figure 4.6: Standard Deviation

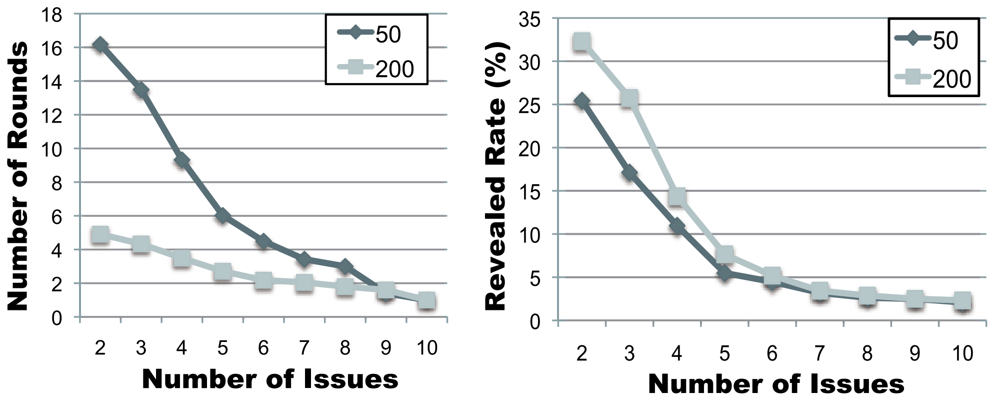

Figure 4.7: Number of rounds

5

Representative based Protocol

5.1 Introduction

Existing works have two main issues. **1) Privacy:** Existing works have not yet addressed agents' private information, which should not be revealed excessively because agents who reveal too much utility information suffer a disadvantage. For example, suppose that several companies are collaboratively designing and developing a new car model. If one company reveals more utility information than the other companies, those other companies can learn more of that company's utility information. As a result, the company will face a disadvantage in subsequent negotiations. Furthermore, explicitly revealing utility information is dangerous from a security standpoint. **2) Scalability:** For example, the basic bidding protocol does not have high scalability for the number of agents. In the basic bidding negotiation protocol, the mediator needs to find the optimum combination of submitted bids from the agents. However, the computational complexity for finding solutions is too large. The number of agent bids was limited in existing work [56]. Limiting bids causes low optimality and high failure rate for agreements.

In this chapter, we propose a representative based protocol that has high scalability for the number of agents and considers the agent's private information. In our protocol, we first select representatives who revealed more of their utility space than the others. These representatives reached an agreement on alternatives and proposed them to the other agents. Finally, the other agents can express their own intentions concerning agreement or disagreement. In this protocol, agents who revealed more private utility information can have a greater chance to be representatives who will attend to reach an agreement on behalf of the other agents. Although agents tend to avoid revealing their own private information, they have an incentive to reveal it to be representatives.

The representative based protocol has been inspired by the parliamentary systems in England, Canada, Australia, Japan, etc. in which representatives are making an agreement on behalf of other people. In a situation in which many people have to reach an agreement, directly reflecting all members' opinions is quite difficult. Doing so requires much time and energy and is not scalable. Although voting is one option, voting might have paradoxical results [4].

We expand our mechanism to be multi-round by using the Threshold Adjustment Protocol in chapter 4. The multi-round mechanism improves the failure rates and achieves fairness in terms of the revealed area. This means that the amounts of the revealed areas are almost the same among agents. Further, a representative mechanism can prevent the unfair solutions that can exist in the original Threshold Adjustment Protocol.

The representative based protocol drastically reduces computational complexity because only representative agents try to reach a consensus. The experimental results demonstrate that our protocol reduces the failure rate in making agreements and that it is scalable on the number of agents compared with existing approaches. We also demonstrate that our protocol reduces the revealed area compared with existing works. Furthermore, we investigate the detailed effect of the representative selection method in our protocol and call the selection method RAS in which agents who reveal a larger utility area are selected as representatives. In the experiments, we compare RAS with a selection method in which representative agents are randomly selected (RANDOM).

The remainder of this chapter is organized as follows. In 5.2, we describe a model of non-linear multi-issue negotiation and an existing work's [56] problems. In 5.3, we propose our new negotiation mechanism. In 5.4, we present an experimental assessment of this protocol. In 5.5, we draw conclusions.

Num. of agents	Limit of bids	Num. of agents	Limit of bids
2	2530	7	9
3	186	8	7
4	50	9	6
5	23	10	5
6	13		

Table 5.1: Limitation of the bids

5.2 Scalability and Privacy Problems

In the basic bidding protocol [56] in chapter 3, agents reach an agreement based on the following steps: Generating bids and Winner Determination. In the winner determination step, the mediator identifies the final contract by finding all the combinations of bids, one from each agent, that are mutually consistent, *i.e.*, that specify overlapping contract regions.

Since it is a combinatorial optimization calculation, computational complexity for finding solutions exponentially increases based on the number of bids. For example, if there are 10 agents and each agent has 20 bids, the number of bids is 20^{10}. To make our negotiation mechanism scalable, the computational complexity must be reduced to find solutions.

The basic bidding protocol [56] were limited the number of bids for each agent to handle the computational complexity. The concrete number of bids in this limitation was $\sqrt[N]{6,400,000}$, a number that reflects our experimental calibration in 2005. But even though CPUs are faster now, the limitation number does not differ so much because this is an exponential problem. Table 5.1 shows the limitation numbers of bids in one agent. This number quickly drops by increasing the total number of agents. Because of the limitation of bids, the failure rate for finding agreements quickly increases along with increasing the number of agents. When the number of agents is five and the number of issues is seven, we experimentally observed that the failure rate is around 40%. In fact, a strong trade-off exists between increasing the number of total bids and finding good

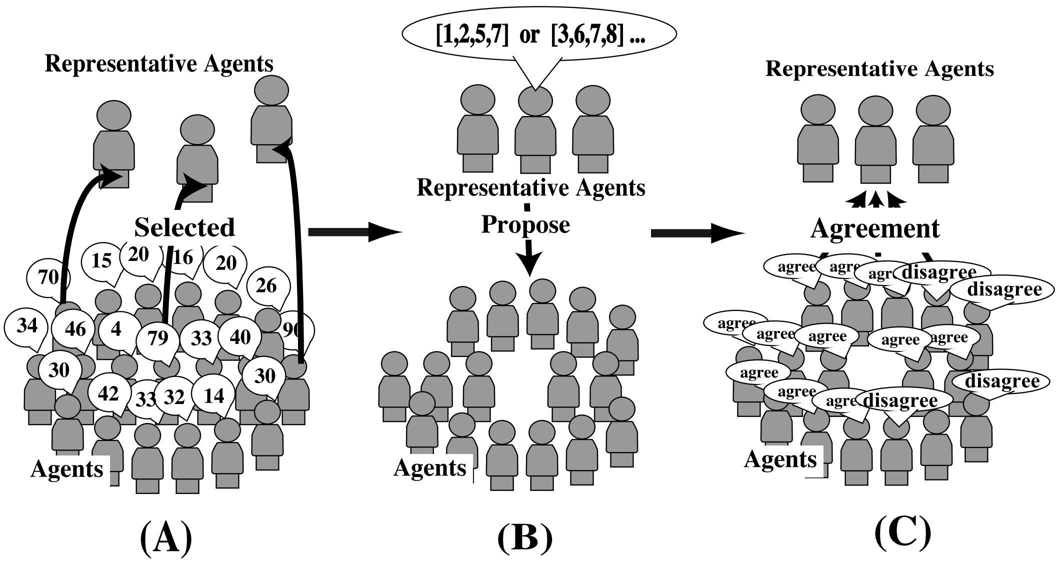

Figure 5.1: Representative based protocol

quality solutions. Increasing the number of total bids is not an effective approach for finding good quality agreements.

Thus, it is necessary to build another mechanism that will find higher quality solutions without limiting the bids. Our mechanism proposed in this chapter is highly scalable. The other issue with existing protocols is that they are not concerned with privacy in utility spaces. Even in a collaborative situation among people, it is normal to keep one's own utility space closed as long as one is not asked to do otherwise. Our new mechanism achieves such a situation by employing the **revealed area** in utility spaces.

5.3 Multi-Round Representative based Protocol based on Revealed Private Information

5.3.1 Representative based Protocol

Representative-based protocol consists of three steps. The first step is to select the representative agents (Step1). The second step is to find solutions, and propose them to the other agents (Step2). The third step is to respond to the agreement by the other agents (Step3).

We assume each agent uses a reservation value to determine whether to "agree" or "disagree" with the representative agents. Actually, for practical applications, the reservation value can be determined by a human user. In addition, we assume that the number of representatives is static in representative based protocol. This protocol consists of the following steps.

[**Step 1: Selection of Representative Agents**] Representative agents are selected based on the amount of their revealed areas, as shown in Figure 5.1 (A). First, each agent submits how much he can reveal his utility space to the mediator. Namely, each agent submits the numeric value of the amount of his possible revealed area. The mediator selects the representative agents who revealed a large area. We call this selection method RAS. This step is main additional coordination processes by the use of representatives. By employing RAS, all agents are satisfied with the mediator's decision because it is the best method for all agents to find optimal solutions.

[**Step 2: Proposing by Representatives**] Representative agents find solutions and propose them to other agents, as shown in Figure 5.1 (B). First, representative agents find solutions by employing a breadth-first search with branch cutting to find solutions (from lines 3 to 14 in representative_protocol()).

Next, the representative agents ask the other agents whether they "agree" or "disagree." Step 2 is repeated until all the other agents agree or the solutions found by the representatives are rejected by the other agents.

[**Step 3: Respond to Agreement by other Agents**] First, the other agents receive the solutions from the representative agents. Then they judge whether they "agree" or "disagree" by determining whether the solution's utility is higher than their own reservation value (Figure 5.1 (C).).

Steps 1, 2, and 3 are captured as Algorithms 2 and 3.

This protocol is scalable for the number of agents. In a representative protocol, combinatorial optimization only occurs in negotiation among representative agents. In fact, the computational complexity for proposing solutions to unrepresentative agents only increases linearly and is almost negligible. Thus,

Algorithm 2 representative_protocol(B)

B: A set of bid-set of each agent
(B=$\{B_0, B_1, ..., B_n\}$, a set of bids from agent i is $B_i = \{b_{i,0}, b_{i,1}, ..., b_{i,m_i}\}$)
RB:A set of bid-set of each representative agent
(RB=$\{RB_0, RB_1, ..., RB_m\}$, a set of bids from representative agent i is $RB_i = \{rb_{i,0}, rb_{i,1}, ..., rb_{i,l_i}\}$)
SC: A set of solution-set of each representative agent
(SC=$\{SC_0, SC_1, ..., SC_n\}$, a set of bids from agent i is $SC_i = \{sc_{i,0}, sc_{i,1}, ..., sc_{i,m_i}\}$)

```
RB := select_representative(B)
SC := RB_0, i := 1
while i < the number of representative agents do
   SC' := ∅
   for s ∈ SC do
      for rb_{i,j} ∈ RB_i do
         s' := s ∪ rb_{i,j}
      end for
   end for
   if s' is consistent then
      SC' := SC' ∪ s'
   end if
   SC := SC', i := i + 1
end while
while i < |SC| do
   if ask_agent(SC_i) is true & SC_i Utility is maximum then
      return SC_i
   else
      return No Solution
   end if
end while
```

Algorithm 3 ask_agent(SC)

select_representative() is a method for performing Step 1

Th: A reservation value of each agent (Th=$\{Th_0, Th_1, ..., Th_n\}$)

```
while i < the number of agents do
  if SC'sUtility < Th_i then
    return false
  else
    i := i + 1
  end if
end while
return true
```

the computational complexity is drastically reduced compared with the existing mechanism.

Finally, we describe the trade-off for an agent between revealing a large amount of utility space and being a representative agent. Representative agents have advantages since they can propose alternatives to other agents and disadvantages because they need to reveal larger utility space. Unrepresentative agents have advantages in keeping their utility hidden and disadvantages in responding based on representatives' agreements.

5.3.2 Multi-Round Representative based Protocol

We extend our protocol to multi-round negotiation based on the threshold adjusting method [35] so that the number of times to be a representative agent is fair. The total amount of revealed utility space for each agent is almost the same by the threshold adjustment mechanism.

The main idea of the threshold adjusting mechanism is simple: if an agent reveals a larger area of his utility space, he should gain an advantage. On the other hand, an agent who reveals a smaller area of his utility space should adjust his threshold to agree with others. The threshold values are changed by each agent based on the amount of revealed area. If the agent decreases the threshold value, this means that he must reveal more of his utility space.

This mechanism is repeated until an agreement is achieved or all agents refuse to lower their thresholds. Agents can decide whether to lower the threshold based on their reservation value, i.e., the minimum threshold. This means that agents have the right to reject the request to decrease their threshold if the request decreases a threshold lower than the reservation value.

The threshold adjusting mechanism is shown as Algorithm 4:

Algorithm 4 threshold_adjustment( )

Ar: Area Range of each agent ($Ar = \{Ar_0, Ar_1, ..., Ar_n\}$)

representative_protocol(): representative based protocol explained in previous section.

loop
    $i := 1$, $B := \emptyset$
    while $i < |Ag|$ **do**
        bid_generation_with_SA(Th_i,V,SN,T,B_i)
    end while
    $maxSolution$:= representative_protocol(B)
    if find $maxSolution$ **then**
        break loop
    else if all agent can lower the threshold **then**
        $i := 1$
        $SumAr := \Sigma_{i \in |Ag|} Ar_i$
        while $i < |Ag|$ **do**
            $Th_i := Th_i - C * (SumAr - Ar_i)/SumAr$
            $i := i + 1$
        end while
    else
        break loop
    end if
end loop
return $maxSolution$

In the threshold adjusting mechanism, agents can consider others' behaviors by adjusting the agent thresholds. In our definition, agents can reveal more revealed area if they greatly lower their threshold. Additionally, the width of decreasing the threshold is decided based on a comparison of the others' revealed areas in the threshold adjusting mechanism. Therefore, they can take the behaviors of others into consideration in multi-round negotiation.

5.4 Experiment Results

5.4.1 Experiment Settings

We conducted several experiments to evaluate the effectiveness of our approach. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions.

In the experiments on optimality, for each run, we applied an optimizer to the sum of all the agents' utility functions to find the contract with the highest possible social welfare. This value was used to assess the efficiency (*i.e.*, how closely optimal social welfare was approached) of the negotiation protocols. To find the optimum contract, we used simulated annealing (SA) because exhaustive search became intractable as the number of issues grew too large. The SA initial temperature was 50.0 and decreased linearly to 0 over the course of 2500 iterations. The initial contract for each SA run was randomly selected.

In terms of privacy, the measurement is the range of the revealed areas. If an agent reveals one point on the grid of the utility space, he loses 1 privacy unit. If he reveals 1000 points, then he loses 1000 privacy units.

We also analyze the representative selection method in our protocol. The representative selection method remains an important research point. The selection method in which agents who reveal a larger utility area are selected as representatives is called RAS, and the random selection method in which representatives are randomly selected is called RANDOM. To investigate the detailed effects of RAS, we assume RANDOM is the general basis for comparison.

The following are the parameters for our experiments:

- Domain for issue values: $[0, 9]$.

- Constraints: 10 unary constraints, 5 binary constraints, 5 trinary constraints, etc. (a unary constraint relates to one issue, a binary constraint relates to two issues, and so on).

- Maximum constraint value: $100 \times (number\ of\ issues)$. Constraints that satisfy many issues have on average larger weights. This seems reasonable for many domains. To schedule meetings, for example, higher order constraints concern more people than lower order constraints, so they are more important.

- Maximum constraint width: 7. The following constraints, therefore, are all valid: issue $1 = [2, 6]$, issue $3 = [2, 9]$ and issue $7 = [1, 3]$.

- Number of samples taken during random sampling: $(number\ of\ issues) \times 200$.

- Annealing schedule for sample adjustment: initial temperature 30, 30 iterations. Note that the annealer must not run too long or too 'hot' because then each sample will tend to find the global optimum instead of the peak of the optimum nearest the sampling point.

- Threshold used by agents to select what to bid starts with 900 and decreases until 200 in the threshold adjusting mechanism. The protocol without the threshold adjusting process defines the threshold as 200. The threshold is used to excise contract points with low utility.

- Amount of threshold is decreased by $100 \times (SumAr - Ar_i)/SumAr$. $SumAr$ means the sum of all agents' revealed areas. Ar_i means agent i's revealed area.

- Limitation on number of bids per agent: $\sqrt[N]{6,400,000}$ for N agents. It was only practical to run the deal identification algorithm if it explored no more than about 6,400,000 bid combinations, which implies a limit of $\sqrt[N]{6,400,000}$ bids per agent for N agents.

- Number of representative agents is two in the representative based protocol.

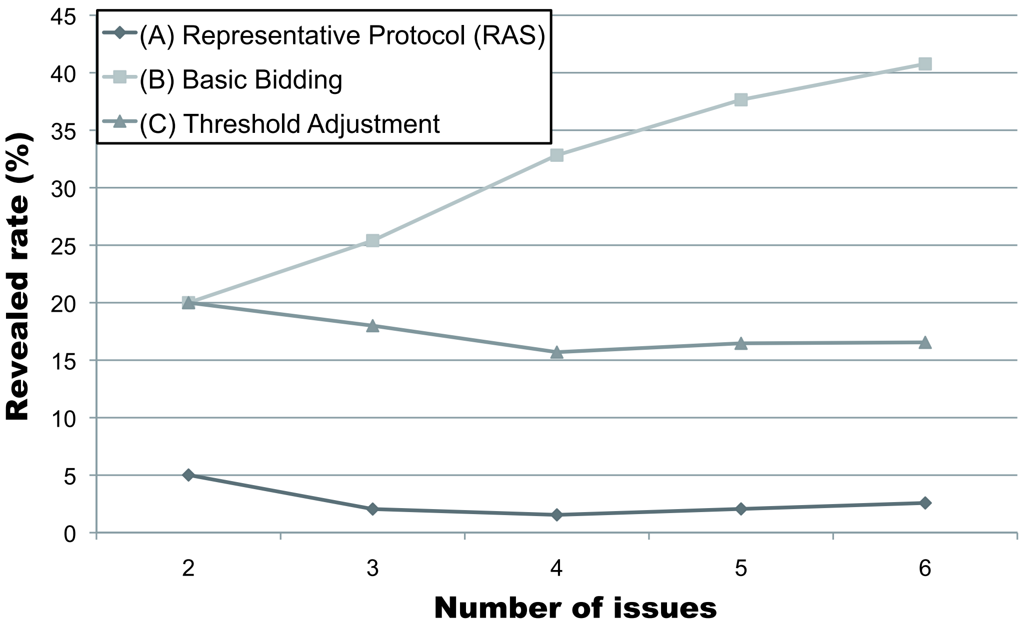

Figure 5.2: Revealed rate

- Number of issues is three.

In our experiments, we ran 100 negotiations in every condition. Our code was implemented in Java 2 (1.5) and run on a core 2 duo processor iMac with 1.0 GB memory on a Mac OS X 10.4 operating system.

5.4.2 Experimental Results

Figure 5.2 shows the revealed rate of three comparable protocols with three agents. (A) is the proposed protocol, which is a multi-round negotiation with the representative protocol whose selection method is RAS. (B) is the basic bidding protocol without threshold adjustment (explained in Section 2). (C) is the protocol with threshold adjustment.

In (B), the revealed rate increases as the number of issues increases. This means that if we do not use the threshold adjustment, agents need to reveal more of their utility space than the other protocols. On the other hand, in (A)

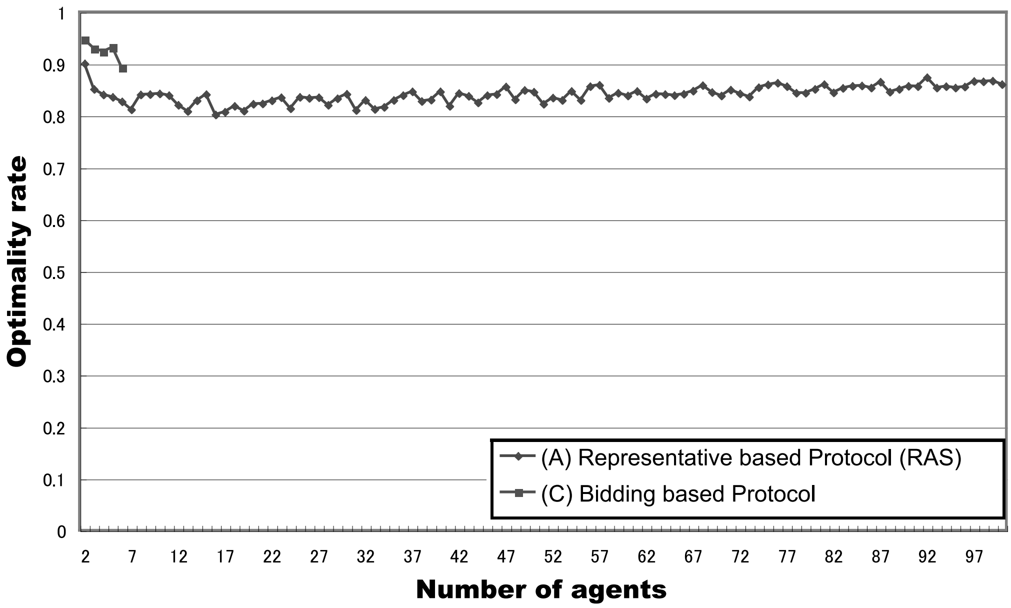

Figure 5.3: Scalability on number of agents

and (C), the revealed rate decreases as the number of issues increases. When we compare (A) with (C), the revealed rate of the representative based protocol is less than basic bidding protocol with threshold adjustment for two reasons. First, the representative protocol finds solutions faster than the threshold adjustment mechanism. Second, basic bidding protocol with threshold adjustment most agents need to reveal their utility space. On the other hand, only representative agents reveal their utility spaces in representative based protocol. Essentially, the representative protocol proposed in this chapter drastically decreases the revealed rate compared with the other two protocols.

The next experimental results show that our negotiation protocol is sufficiently scalable on the number of agents. Figure 5.3 shows optimality when the number of agents ranges from 2 to 100. In this experiment, we assume agents have a shared utility area that is agreeable for them. This is because when the number of agents becomes large, finding an agreement point is quite hard using negotiation protocols and comparing optimality could be impossible. To create a common area, agents' utility spaces are randomly generated. Then a common area is

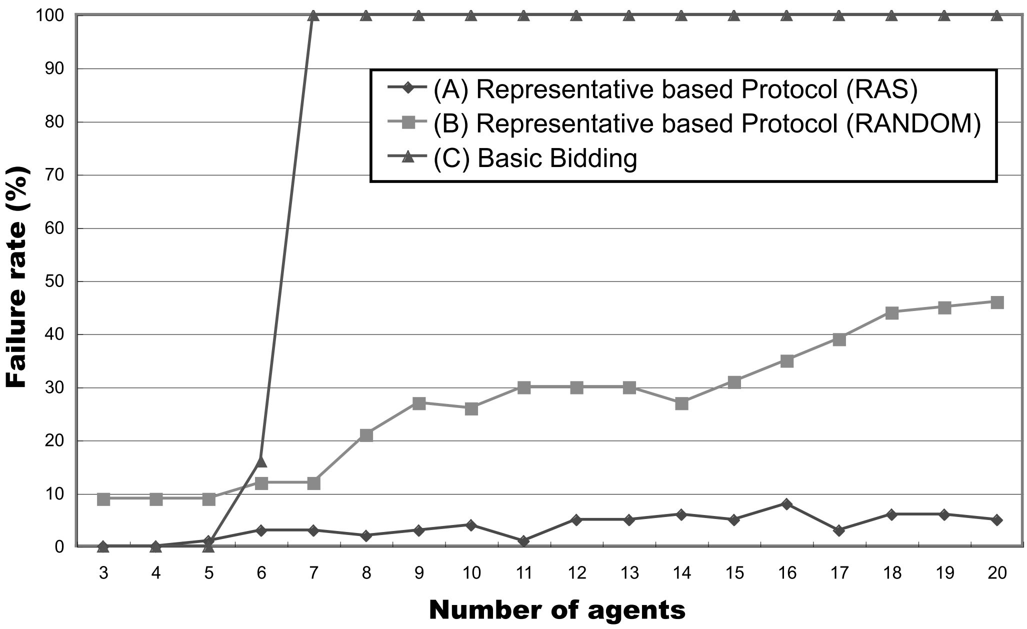

Figure 5.4: Failure rate

randomly generated whose value is more than an agent's threshold.

The results demonstrated that optimality is more than 80% in all cases in spite of not finding solutions to 7 issues in existing work. Although high optimality came from the above common area assumption, the scalability of our new protocol is ensured by this experiment. Our proposed approach works well in single issue negotiations and multiple independent issues negotiations as well because such negotiations have lower computational complexity than multiple interdependent issues negotiations.

Figure 5.4 shows the failure rate for finding solutions in the three protocols. (A) is the representative based protocol and selection method RAS. (B) is the representative based protocol and selection method RANDOM. (C) is the basic bidding protocol without threshold adjustment explained in Section 2. Even if the number of agents increases, (A) is almost 0. On the other hand, (C) shows a drastic increase over five agents because the bid limitation starts when there are five agents. Also, for more than five agents, the existing mechanism fails to

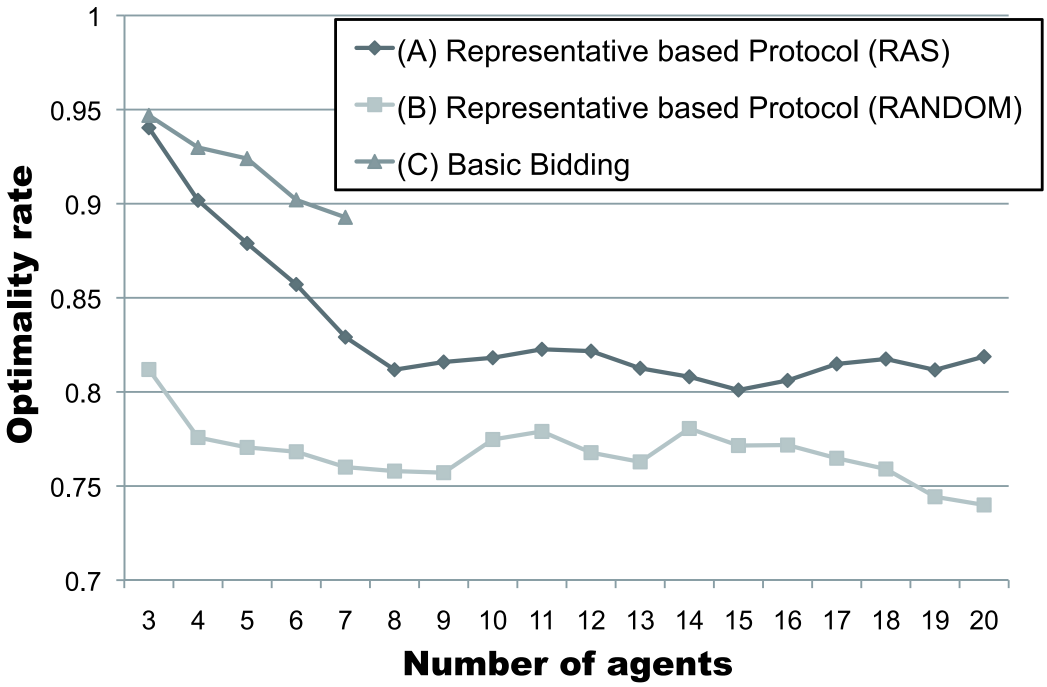

Figure 5.5: Comparisons of optimality

find solutions. Furthermore, (A) and (B) show that RAS has a lower failure rate than RANDOM. Thus, the representative protocol with selection method RAS has better failure rates.

Figure 5.5 compares optimality rates among "(A) Representative based Protocol (RAS)," "(B) Representative based Protocol (RANDOM)," and "(C) Basic Bidding." Comparing (A) and (C), the difference of optimality is small, around 0.05 at most. This difference reflects that since the representative based protocol tends to find solutions at an early stage, it might miss better solutions. Furthermore, (A) and (B) show that RAS has higher optimality than RANDOM because more solutions are found in representatives with large revealed areas. Thus, the representative protocol with selection method RAS has better optimality rates

Figure 5.6 shows the variance of the utility value per agent. By this experiment, we can recognize the satisfactory rate of individual agents. The variance of the utility per agent is critical in bargaining theory because some experimen-

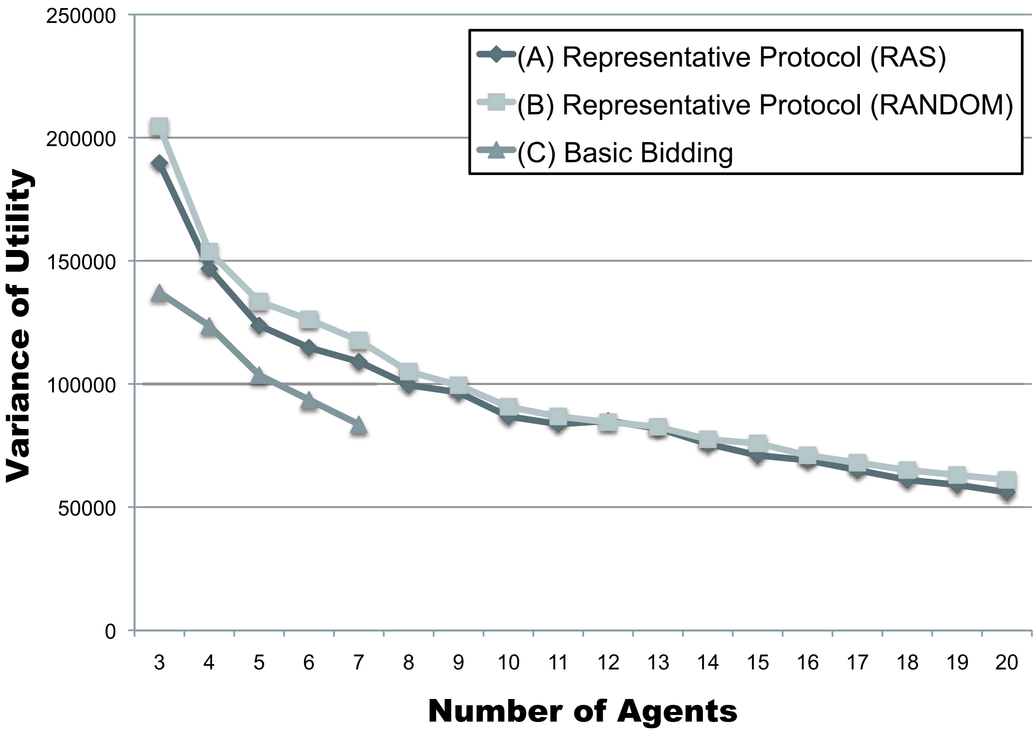

Figure 5.6: Comparison on variance of utility

tal results suggest that fairness influences decision-making per agent ([148] etc.). (C) is better than (A) because the utility of the representatives is higher than that of the unrepresentatives in the representative protocols. However, all agents definitely satisfy the agreement points because their utility values are higher than their reservation value.

Figure 5.7 shows the optimality and failure rates on the number of representative agents. In this experiment, there are seven agents and three issues. Figure 5.7 (i) shows that optimality increases when the number of representative agents increases. Even if agents have low utilities, they tend to be persuaded by representative agents when the number of representatives is small. Figure 5.7 (ii) shows that the failure rate sharply increases when the number of representative agents exceeds, which is where the bid limitation starts.

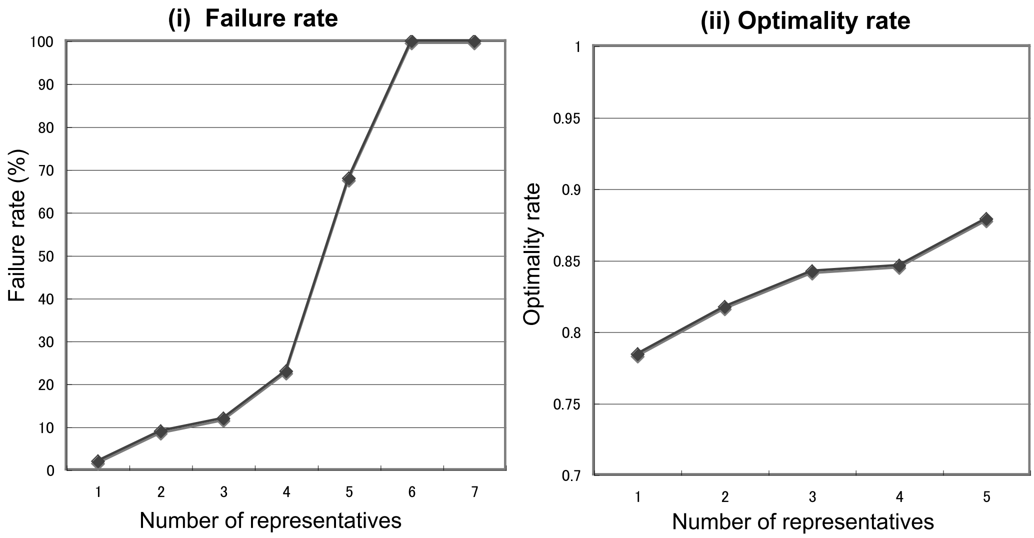

Figure 5.7: Optimality rate and failure rate in number of representative agents

5.5 Conclusion

In this chapter, we proposed a multi-round representative based protocol in very complex negotiations among software agents. The representative based protocol always reached agreements if the number of agents was large. It is important for agents to make agreements without revealing their private information during the negotiations. This proposed protocol reached an agreement while revealing as little agents' utility space as possible. The experimental results demonstrated that the representative based protocol reduced the amount of private information required for an agreement among agents, and its failure rate was almost 0. Furthermore, we compared RAS with RANDOM in the experiments. The failure rate in RAS was lower than RANDOM.