6

Secure and Efficient Negotiation Protocols for Cube-Constraints and Cone-Constraints

6.1 Introduction

In this chapter, we propose the Distributed Mediator Protocol (DMP) and the Take it or Leave it (TOL) Protocol. They make agreements and conceal agent utility values. In the Distributed Mediator Protocol, we assume many mediators who search in utility space to find agreements. When searching in their search space, they employ the Multi-Party Protocol with which they can simultaneously calculate the sum 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.

We also propose the Hybrid Secure Protocol (HSP) that combines DMP with TOL. In Hybrid Secure Protocol (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. Hybrid Secure Protocol (HSP) can also reach an agreement and conceal per agent utility information. Additionally, Hybrid Secure Protocol

(HSP) can reduce the required memory for making an agreement, which is a major issue in DMP. Moreover, we demonstrate that HSP can improve communication cost (memory usage) more than DMP.

In general, although DMP and HSP are protocols among agents and mediators, they do not define the agreement search method, which means how the mediator searches and finds agreement points. Thus, we examine three agreement search methods, a hill climbing(HC), a simulated annealing(SA) and a genetic algorithm(GA) in cone-constraint and cube-constraint situations. Hill climbing and simulated annealing have been employed in the previous works [56]. However, genetic algorithm also performs well to find high optimal contract. Therefore, we compare GA-based method with the other methods in this chapter.

The remainder of the chapter is organized as follows. In 6.2, we describe utility functions based on cube-constraints and cone-constraints. In 6.3, we propose the Distributed Mediator Protocol (DMP) and the Take it or Leave it (TOL) Protocol. In 6.4, we propose the Hybrid Secure Protocol (HSP). In 6.5, we present the experimental results about optimality and communication cost (memory). In 6.6, we draw conclusions.

6.2 Cube-Constraints and Cone-Constraints

Cube-constraints: An agent's utility function is described in terms of cube-shaped constraints [56]. It was explained in chapter 3.

Cone-constraints: An agent's utility function can be described in terms of cone-constraints. Figure 6.1 shows an example of a binary cone-constraint between Issues 1 and 2. This cone-constraint has a value of 20, which is maximum if the situation is $\vec{s}_{central} = [2, 2]$. The impact region is $\vec{w} = [1, 2]$. The expression for a segment of the base is $(x_1 - 2)^2 + (x_2 - 2)^2/4 = 1$[1].

Suppose there are l cone-constraints, $C = \{c_k | \ 1 \leq k \leq l\}$. Cone-constraint c_k has gradient function $g_k(\vec{s}_{central}, \vec{w})$, which is defined by two values: central value $\vec{s}_{central}$, which is the highest utility in c_k, and impact region $\vec{w}$,

[1]The general expression is $\sum_{i=1}^{m} x_i^2/w_i^2 = 1$

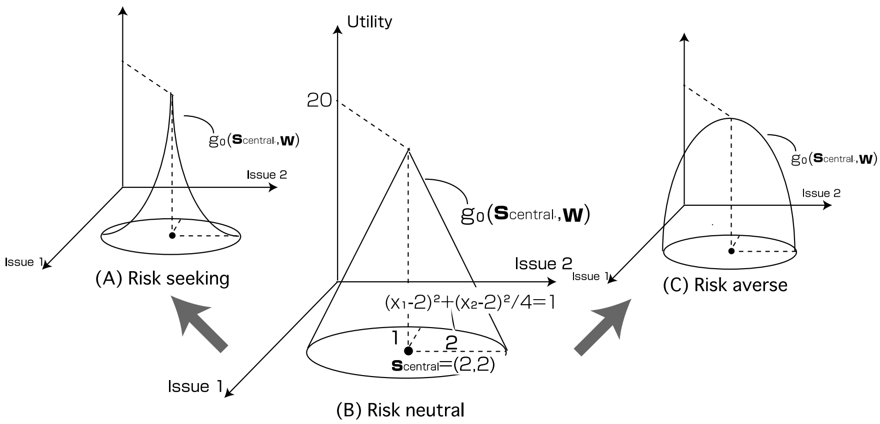

Figure 6.1: Example of cone-constraints

which represents the region where c_k is affected. We assume not only circle-based but also ellipse-based cones. Thus constraint c_k has value $u_i(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}$. Impact region $\vec{w}$ is not a value but a vector. These formulas can represent utility spaces if they are in a n-dimensional space.

In addition, cone-constraints can include the risk attitude for constraints by configuring gradient function $g_k(\vec{s}_{central}, \vec{w})$. This risk means the possibility to fail to make agreements. If the agent usually has a risk neutral attitude for c_k, g_k is defined as (B) in Figure 6.1 (e.g., proportion). However, the attitudes (types) of agent can change from risk-seeking to risk-averse for making agreements. For example, if agents have a risk-seeking attitude for constraint c_k, g_k is defined as (A) in Figure 6.1 (e.g., exponent). If an agent has a risk-averse attitude for c_k, g_k is defined as (C) in Figure 6.1. If agents have the most risk-averse attitude for c_k, g_k stays constant. Therefore, c_k is shaped like a column if the agents have the most risk-averse attitude.

An agent's utility for contract $\vec{s}$ is defined as $u_i(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_i(c_k, \vec{s})$, where $x(c_k)$ is a set of possible contracts (solutions) of c_k. This expression produces a "bumpy" nonlinear utility space with high points where many constraints are satisfied and lower regions where few or no constraints are satisfied.

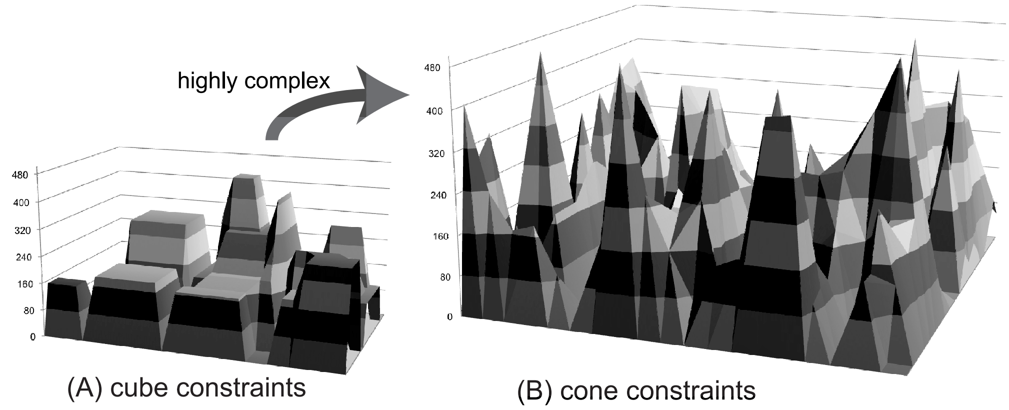

Figure 6.2: Example of utility space

Figure 6.2 shows an example of a nonlinear utility space with two issues. This utility space is highly nonlinear with many hills and valleys. Compared with cube-constraints, the utility function is highly complex because its highest point is narrower. Therefore, the protocols for making agreements must search in highly complex utility space. A simple simulated annealing method to directly find optimal contracts is especially insufficient in a utility function based on cone-constraints.

We assume, as is common in negotiation contexts, that agents do not share their utility functions with each other to preserve a competitive edge. Generally, in fact, agents do not completely know their desirable contracts in advance, because their 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, which is too many to evaluate exhaustively. Agents must thus operate in a highly uncertain environment.

6.3 Secure Negotiation Protocol

6.3.1 Distributed Mediator Protocol (DMP)

We propose the Distributed Mediator Protocol (DMP) in this subsection. We assume there are more than two mediators (**Distributed Mediator**) so that DMP achieves distributed search and protection of the agent's private information by employing the Multi-Party Protocol[87, 131]. DMP is shown as follows.

We assume m mediators $(M_0, \ldots, M_m)$ who can calculate the sum of all the agent utility values if k mediators get together, and there are n agents $(Ag_0, \ldots, Ag_m)$. All mediators share q, which is preliminarily the prime number.

Step 1: The mediators divide the utility space (search space) and choose a mediator who manages it. How to divide the search space and assign tasks is beyond the scope of this discussion. Parallel computation is possible by dividing the search space. This means that the computational complexities during searching can decrease.

Step 2: Each mediator searches his/her search space with a local search algorithm [123]. Hill-climbing search (HC) and simulated annealing search are examples of local search algorithms. The objective function using a local search algorithm is used to maximize the social welfare. During the search, the mediator declares a Multi-Party Protocol if he/she is searching in the state for the first time. After that, the mediator selects k mediators from all mediators and asks for generating v(shares) from all agents.

Step 3: Agent i (A_i) randomly selects k dimension formula, which fulfills $f_i(0) = x_i$, and calculates $v_{i,j} = f_i(j)$. (x_i: agent's i's utility value). After that, A_i sends $v_{i,j}$ to M_j.

Step 4: Mediator j (M_j) receives $v_{1,j}, \ldots, v_{n,j}$ from all agents. M_j calculates $v_j = v_{1,j} + \cdots + v_{n,j} \ mod \ q$ and reveals v_j to the other mediators.

Step 5: The mediators calculate $f(j)$, which fulfills $f(j) = v_j$ by Lagrange's interpolating polynomial. Finally, s, which fulfills $f(0) = s$, is the sum of all agent utility values.

Steps 2 $\sim$ 5 are repeated until they fulfill the at-end condition in the local search algorithm.

Step 6: Each mediator informs the maximum value (alternative) in his space to all mediators. After that, the mediators select the maximum value from all alternatives.

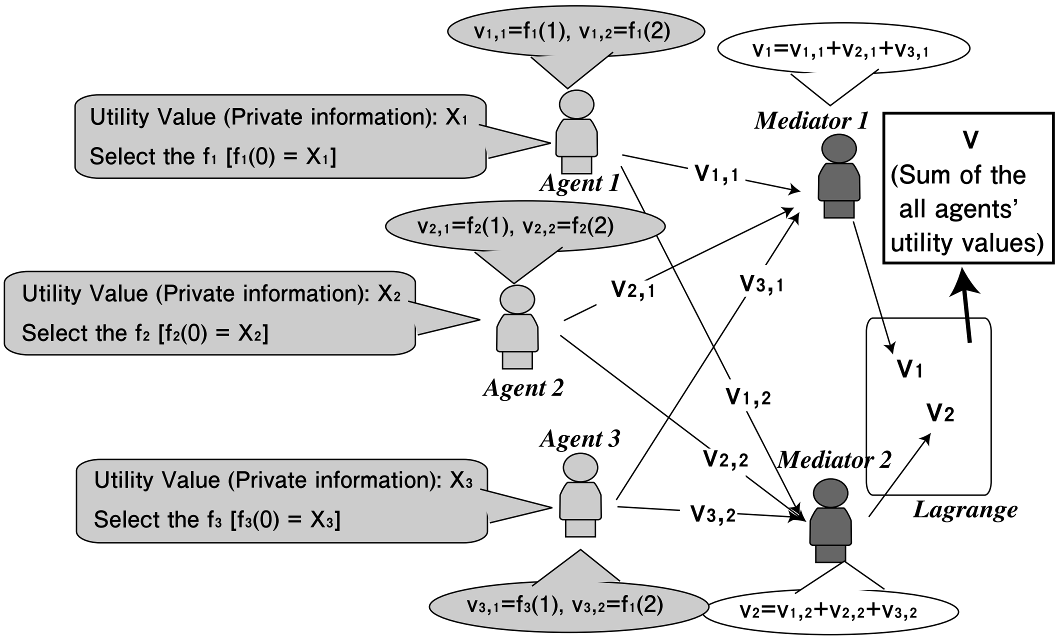

Figure 6.3: Distributed Mediator Protocol

Figure 6.3 shows the flow in DMP. There are three agents and two mediators. If two mediators get together, they can calculate the sum of the per agent utility value. The gray area shows that agents perform the steps without revealing them. As the figure indicates, the selection of multinomial (f_i), generating share (v), adding the share, and Lagrange's interpolating polynomial can calculate the sum of all agent utility values and conceal them.

DMP has an advantage for privacy for an agent's utility information and scalability for utility space. The details are shown as follows.

Privacy DMP can calculate the sum of all agent utility values and conceal them. The proof is identical to the Multi-Party Protocol [87]. In DMP, other agents and the mediators can't know the utility values without illegally colluding.

Additionally, k, which is the number of mediators performing the Multi-party protocol, is the tradeoff between privacy issues and computational complexity. If k mediators exchange their shares (v) illegally, they can expose the agent utility values. Therefore, it is good for protecting an agent's privacy information that k is large number that mediators can't collude illegally. If k is large number, mediators take a lot of trouble with colluding illegally. However, it requires more computation time because more mediators have to stop searching.

Scalability The computational cost can be greatly reduced because the mediators divide the search space. In existing protocols, they cannot find better agreements when the search space becomes too large. However, this protocol can locate better agreements in large search spaces by dividing the search space.

DMP has a weak point: too many shares (v) are generated. This is because shares are generated that correspond to the search space. To generate shares requires much more communication cost with agents than searching without generating shares. Thus, we need to generate fewer shares with high optimality.

6.3.2 Take it or Leave it (TOL) Protocol for Negotiation

We propose the *Take it or Leave it (TOL) Protocol*, which can also reach agreements and conceal all agents' utility information. The mediator searches with the hill-climbing search algorithm [123], which is a simple loop that continuously moves in the direction of increasing evaluated value. Values for each contract is evaluated by the responses that agents take or leave to the offers to move from the current state to the neighbor state. The agents can conceal their utility value using this evaluation value. This protocol consists of the following steps.

Step 1: The mediator randomly selects the initial state.

Step 2: The mediator asks the agents to move from the current to the neighbor state.

Step 3: Each agent compares its current state with the neighbor state and determines whether to take or leave it. If the neighbor state provides higher utility value than the current state, the agent "takes it." If the current state provides higher or identical utility value than the neighbor state, the agent "leaves it."

Step 4: The mediator selects the next state declared by the most agents as "take it." However, the mediator selects the next state randomly if there are more than two states that most agents declared as "take it." The mediator can prevent the local maxima from being reached by random selection.

Steps 2, 3, and 4 are repeated until all agents declare "leave it" or the mediator determines that a plateau has been reached. A plateau is an area of the state space landscape where the evaluation function is flat.

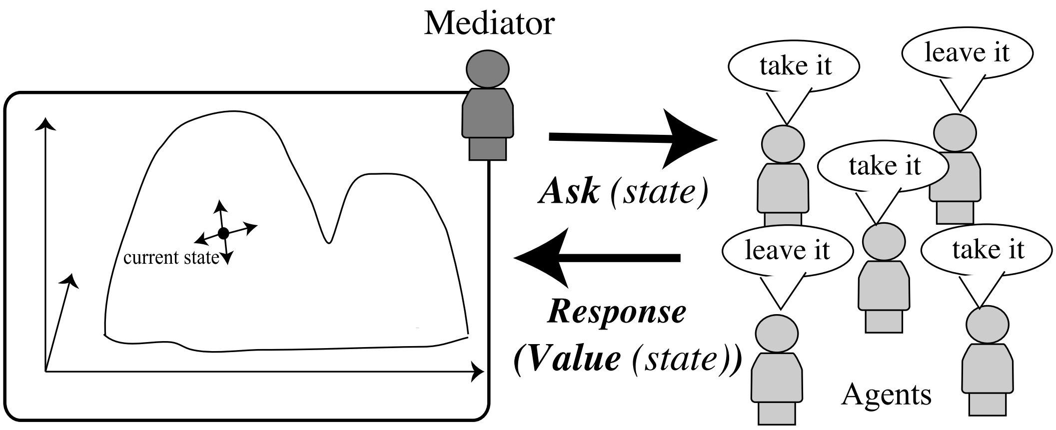

Figure 6.4: Take it or Leave it (TOL) Protocol

Figure 6.4 shows the concept of the "Take it or Leave it (TOL) Protocol." First, the mediator informs agents about the state whose evaluation value he wants to know. Second, agents search for their utility space and declare "take it" or "leave it." Then they tell the number of agents who declare "take it" (*VALUE (state)*). These steps are repeated until they satisfy the at-end condition.

"Take it or Leave it (TOL) Protocol" has an advantage of lower time complexity because it easily rates evaluated value. However, this protocol can't find high optimality solutions when a plateau is reached.

6.4 Hybrid Secure Negotiation Protocol (HSP)

We propose a new protocol that combines DMP with TOL to solve DMP's weak point. This new protocol is called the Hybrid Secure Protocol (HSP) for negotiation. HSP generates fewer shares than DMP. The Hybrid Secure Protocol (HSP) is shown as follows.

Step 1: The mediators divide the utility space (search space) and choose a mediator who manages it.

Step 2: Each mediator searches in her search space using TOL proposed in 6.3.2. The initial state is selected randomly. By performing the TOL at first, the mediators can find somewhat higher optimality of solutions without generating shares (v).

Step 3: Each mediator searches in her search space using step 2 - step 5 in DMP proposed in 6.3.1. The initial state is the solution found in previous step. By performing DMP after TOL, mediators can find the local optima in the neighborhood and conceal the per agent private information.

Steps 2 and 3 are repeated many times by changing the initial state.

Step 4: Each mediator communicates the maximum value (alternative) in his space to all mediators. After that, the mediators select the maximum value from all alternatives. Finally, the mediators propose this alternative as the agreement point.

HSP can find solutions with fewer shares than DMP because the initial state in Step 3 is higher than only performing DMP. In addition, TOL doesn't generate shares, and DMP searches in states in which TOL hasn't searched. Thus, HSP can reduce the number of shares. Furthermore, TOL and DMP can protect agents' utility value (private value). Therefore, HSP can also protect agents' utility value.

Meanwhile, optimality in HSP is higher. TOL usually stops searching after reaching the plateau. Additionally, the main reason for lowering the optimality in DMP is to reach the local optima, although the initial value in Step 3 is usually different because it is decided by TOL. Therefore, HSP can find higher agreement in optimality.

6.5 Experimental Results

6.5.1 Setting of Experiment

We conducted several experiments to evaluate the effectiveness of our approach. 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. The following are the parameters for our experiments. The number of agents was six, and the number of mediators was four.

We compared the following methods: "(A) DMP (SA)" is the Distributed Mediator Protocol and the search algorithm is simulated-annealing [123]. "(B) DMP (HC)" is the Distributed Mediator Protocol and the search algorithm is hill-climbing [123]. "(C) DMP (GA)" is the Distributed Mediator Protocol and the search algorithm is the genetic algorithm [123]. "(D) HSP (SA)" is the hybrid secure protocol, and the search algorithm in the distributed mediator step is simulated annealing. "(E) HSP (HC)" is the hybrid secure protocol, and the search algorithm in the distributed mediator step is the hill-climbing algorithm.

In the optimality experiments, for each run, we applied an optimizer to the sum of all agent 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, which decreased linearly to 0 over the course of 2500 iterations. The initial contract for each SA run was randomly selected. Optimality rate is defined as *(The maximum utility value calculated by each method) / (Optimum contract value using SA).*

The following are the parameters for our experiments:
The number of agents is six, and the number of mediators is $2^{(the\ number\ of\ issues)}$. In DMP, they can calculate the sum of the per agent utility values if four mediators get together. In DMP, the search space is divided equally.

Utility function (Cube-constraint) The domain for the issue values is $[0, 9]$. Constraints include 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). The value for a constraint is $100 \times (Number\ of\ Issues)$. Constraints that satisfy many issues have, on average, larger weights, which seems reasonable for many domains. To meet scheduling, for example, higher order constraints concern more people than lower order constraints, so they are more important. 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 $7 = [1, 3]$.

Utility function (Cone-constraint) The domain for the issue values, the number of constraints and maximum width for a constraint are similar to the setting of cube-constraints. The maximum value for a constraint is $100 \times (Number\ of\ Issues)$. The gradient function is defined as $u(\vec{s}) = (Max\ Value) * (1 - (distance)/(width))$. ($u(\vec{s})$: utility value at $\vec{s}$ when $\vec{s} is in the cone - constraints$, $(distance)$: distance between $\vec{s}$ and the central point, $(width)$: impact region, *(Max Value)*: value at the central point)

We set the following parameters for the search methods: HC, SA, and GA.

Hill climbing (HC): The number of iterations is *20 + (Number of issues)* $\times$ *5*. The final result is the maximum value achieved.

Simulated annealing (SA): The annealing schedule for the distributed mediator protocol included a initial temperature is 50. For each iteration, the temperature is decreased by 0.1. Thus, it decreased to 0 by 500 iterations. *20 + (Number of issues)* $\times$ *5* searches are conducted while the initial start point is being changed. The annealing schedule for the hybrid secure protocol in distributed mediator protocol step included an initial temperature of 10 with 100 iterations. Note that the annealer must not run too long or too 'hot' because then each initial state by TOL will tend to find the global optimum instead of the peak of the optimum nearest the initial state in DMP.

Genetic algorithm (GA): The population size in one generation is *20 + (Number of Issues)* $\times$ *5*. We employed a basic crossover method in which two parent individuals are combined to produce two children (one-point crossover). The fitness function is the sum of all agents' (declared) utility. 500 iterations were conducted. Mutations happened at very small probability. In a mutation, one of the issues in a contract vector was randomly chosen and changed. In the GA-based method, we define an individual as a contract vector.

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 10.5 operating system.

6.5.2 Experimental Results

Figure 6.5 shows the optimality rate in five protocols in "cube"-constraints situation. "(B) DMP (HC)" decreases rapidly based on the number of issues because hill-climbing reaches local optima by increasing the search space. "(C) DMP (GA)" does not decrease rapidly even if the number of issues increased. Additionally, "(A) DMP (SA)" is the same as the optimal solution. Therefore, optimality in DMP depends on the search algorithm. "(D) HSP (HC)" have high optimality because HSP performs DMP after performing TOL. In addition, "(D) HSP (HC)" has higher optimality than "(C) HSP (SA)" because SA in the DMP

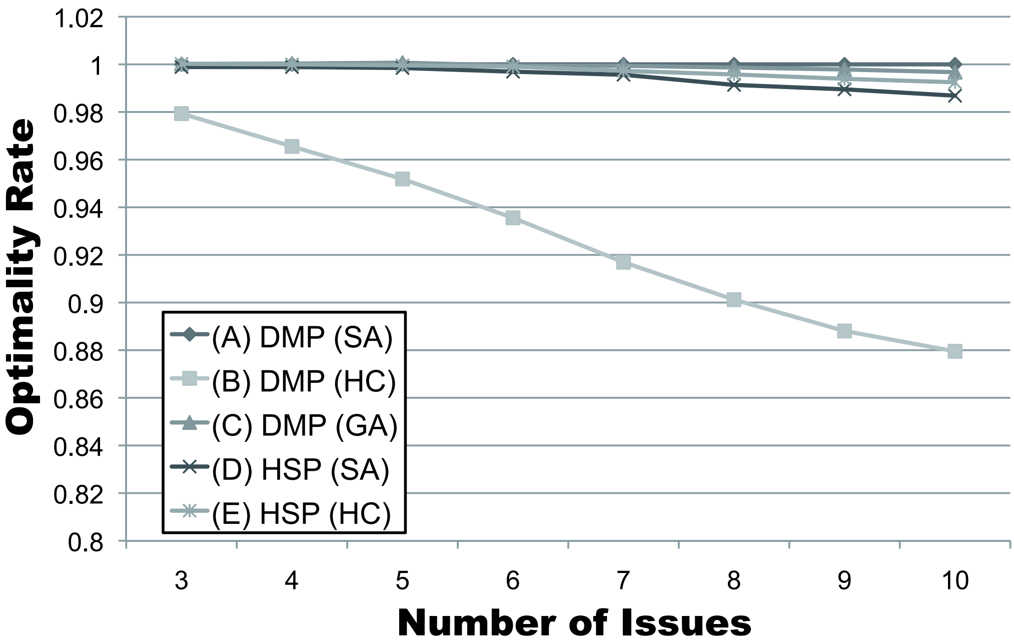

Figure 6.5: Optimality Rate (Cube-constraints)

step sometimes stops searching for a worse state than the initial state due to a random nature. But HC stops searching for a better state than the initial state.

Figure 6.6 shows the optimality rate in five protocols in "cone"-constraints situation. "(B) DMP (HC)" decreases rapidly based on the number of issues and "(C) DMP (GA)" does not decrease rapidly even if the number of issues increased. Therefore, optimality in DMP is similar results in cone-constraints situation. "(D) HSP (HC)" also have high optimality and "(D) HSP (HC)" has higher optimality than "(C) HSP (SA)." Therefore, "(D) HSP (HC)" has high optimality if the utility function is cone-constraints. However, the difference among per protocol in cone-constraints is larger than the one in cube-constraints because the utility space in cone-constraints is more complex.

Figure 6.7 shows the average share (v) per agent in cube-constraints. The number of shares shows a comparison of memory in several protocols. "(C) DMP (GA)" increases exponentially. On the other hand, "(A) DMP (SA)" and "(B) DMP (HC)" reduces the shares compared to "(C) DMP (GA)" because GA

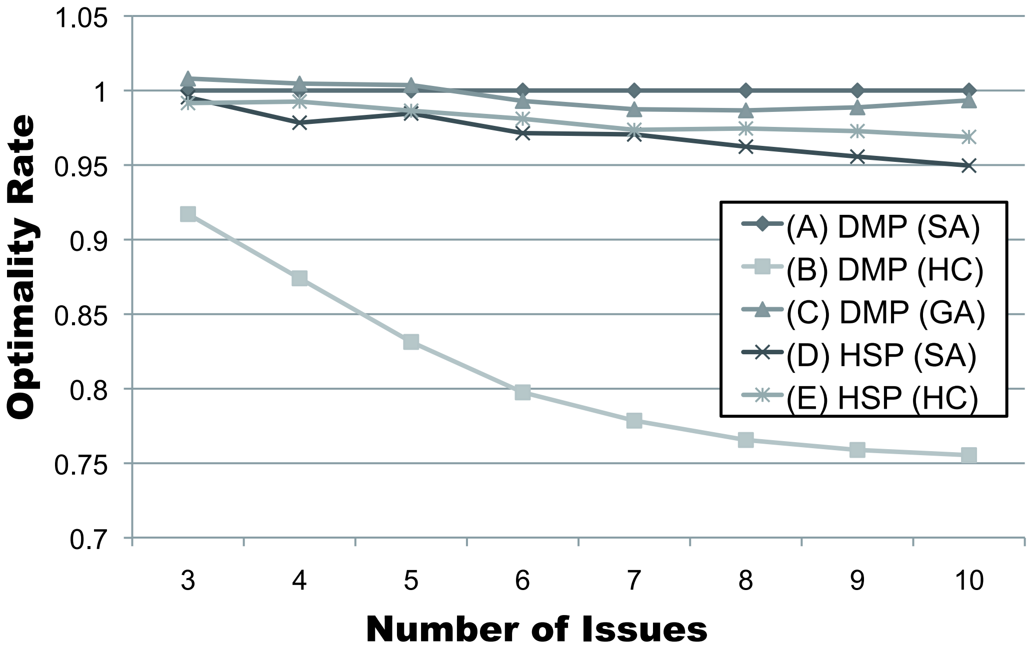

Figure 6.6: Optimality Rate (Cone-constraints)

searches for more states than SA and HC. The number of shares in DMP depends on the features of the search protocol. Furthermore, "(C) HSP (SA)" and "(D) HSP (HC)" reduce shares compared to "(A) DMP (SA)," "(B) DMP (HC)" and "(C) DMP(GA) because the initial state in the DMP step in HSP has a higher value than the initial state in DMP since TOL was performed before. Thus, HSP can reduce the shares more than DMP.

Figure 6.7 shows the average share (v) per agent in cone-constraints. "(C) DMP (GA)" increases exponentially if the utility function is cone-constraints. The number of shares in DMP depends on the features of the search protocol in cone-constraints situation. Furthermore, "(C) HSP (SA)" and "(D) HSP (HC)" also reduce shares compared to "(A) DMP (SA)," "(B) DMP (HC)" and "(C) DMP(GA) in cone-constraint situation. Thus, HSP can reduce the shares more than DMP if the utility function is cone-constraints. The number of shares in cone-constraints situation is overall less than the one in cube-constraints situation. This is because that all search methods in cone-constraints have higher possibility

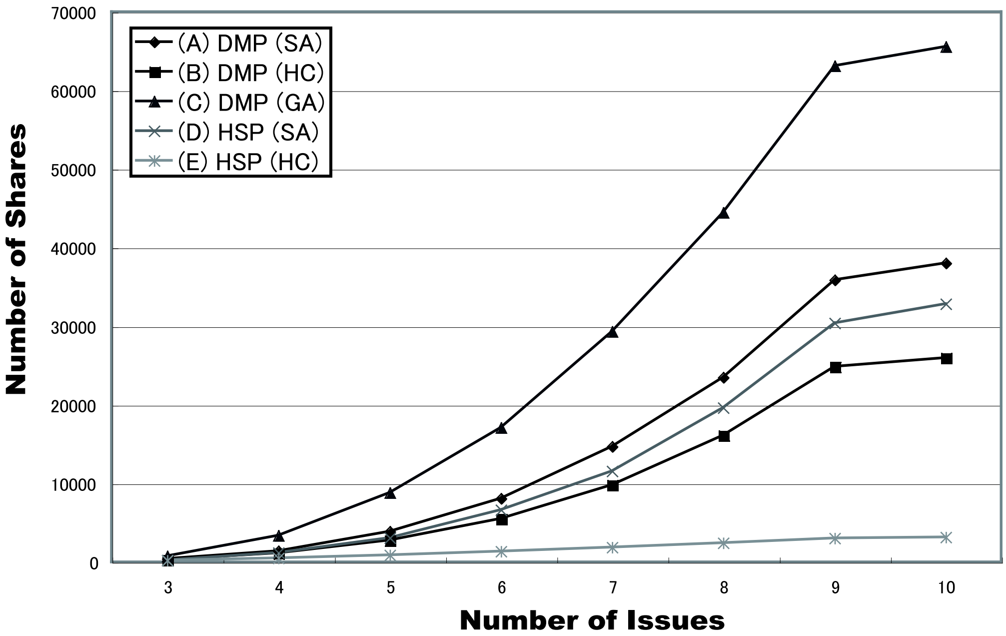

Figure 6.7: The number of shares (Cube-constraints)

to reach local optima due to the utility space's complexity.

From the above experimental results, HSP can reduce the shares with high optimality.

6.6 Conclusion

In this chapter, we proposed a nonlinear utility function based on cone-constraints and proposed the Distributed Mediator Protocol (DMP) that can reach agreements and completely conceal agent's utility information and achieve high scalability in utility space. Moreover, we proposed the Hybrid Secure Protocol (HSP) that combines DMP and Take it or Leave it (TOL) protocol. Experimental results demonstrated that HSP can reduce memory with high optimality in cone-constraints and cube-constraints situations.

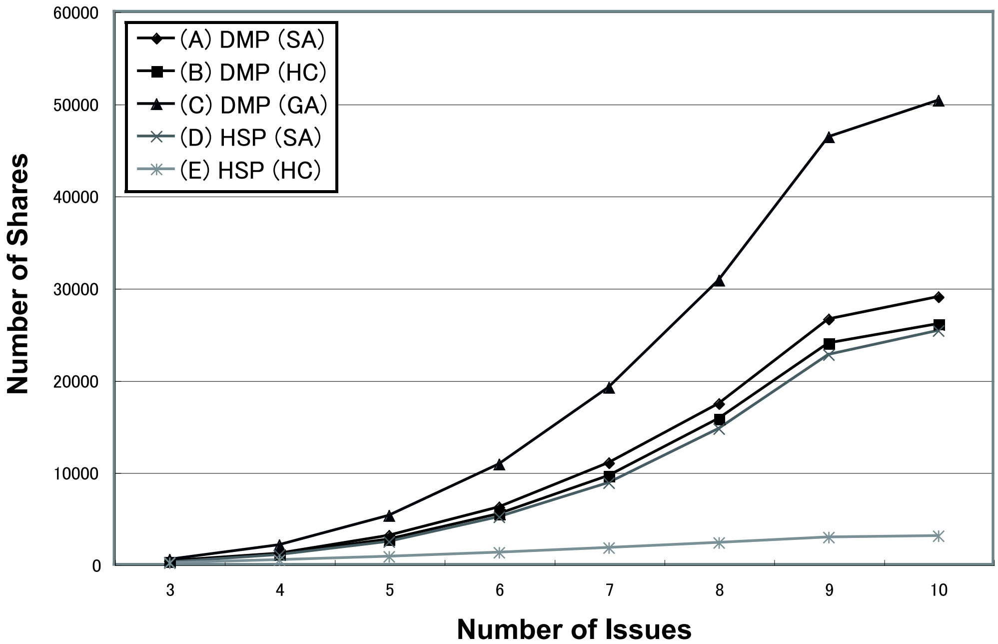

Figure 6.8: The number of shares (Cone-constraints)

7

A Secure and Fair Protocol that Addresses Weaknesses of the Nash Bargaining Solution

7.1 Introduction

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 *non-linear* 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. We need, therefore, a new approach if we want to produce good outcomes for nonlinear negotiations.

In this chapter, we present a secure mediated protocol (the Secure and Fair Mediator Protocol, or SFMP) that addresses this challenge. The protocol consists of two main steps. In the first step, SFMP uses a nonlinear optimizer, integrated with a secure information sharing technique called the Secure Gathering Protocol [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, which we call approximate fairness, that measures how equally the total utility is divided across the negotiating agents ([115] etc.). We demonstrate that SFMP produces better scalability and social welfare values than previous nonlinear negotiation protocols.

The remainder of this chapter is organized as follows. In 7.2, we show how the Nash Bargaining Solution can lead to sub-optimal results in such contexts. In 7.3, we describe a new protocol (SFMP) designed to address this challenge. In 7.4, we present the experimental results. In 7.5, we draw conclusions.

7.2 A Weaknesses of the Nash Bargaining Solution in Nonlinear Negotiation

Working in the nonlinear domain has a number of important impacts on the kind of negotiation protocols that can be effective. First, consider parero-optimality. Pareto-optimality is widely recognized as a basic requirement for a good negotiation outcome. It is defined as follows: Contract $\vec{s} = (s_1, \ldots, s_M)$ is **Pareto optimal** if there is no $\vec{s}'$ such that $u_i(\vec{s}\,') > u_i(\vec{s})$ for all agents ($u_i(\vec{s})$ is agent i's utility value). Pareto-optimality thus eliminates all contracts where there are others that are better for all the parties involved. In a linear negotiation (i.e. where the agent utility functions are defined as the weighted sum of the values for each issue), it is computationally trivial to find the Pareto front, and the social welfare (sum of agent utilities) for every contract on the Pareto frontier is the same. In fact, the Pareto-optimal frontier for a negotiation will be "sparse" in our model, i.e. the Pareto-optimal contracts points will be few in number and widely scattered.

Next, let us consider fairness. Fairness is critical in bargaining theory because some experimental results suggest that it deeply influences human decision-making ([70] etc.) in such contexts as family decision making (e.g., where will we go on our next vacation?), the less formal economy of consumer transactions (such as ticket scalpers or flea markets), and price setting for consumer purchases. The ultimatum game is a popular example of this effect [2, 7]. People tend to offer "fair" (i.e., 50:50) splits, and offers less than 20% are often rejected in this game,

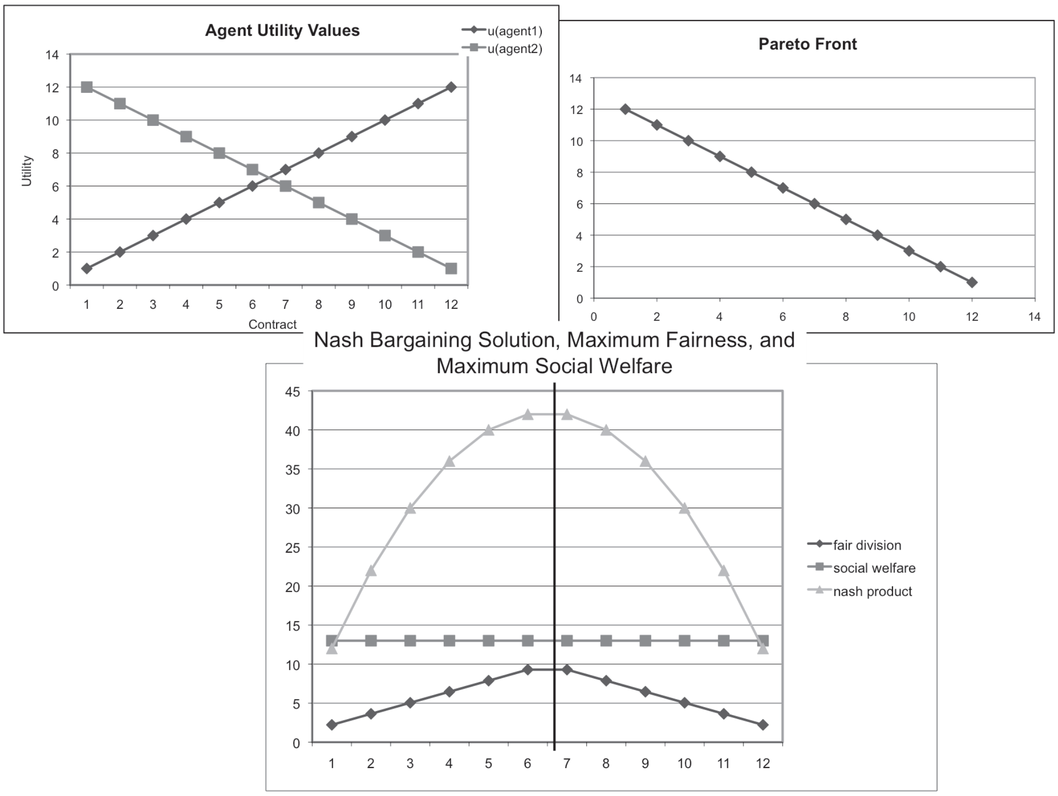

Figure 7.1: The relationship of nash product, fairness and social welfare in a linear utility function

even though it is irrational in this game to reject any deal, since the alternative is a zero payoff. There are many other studies about the relationship between decision making and "fairness" in the experimental economics and behavioral economics fields [69, 148].

The Nash Bargaining Solution (i.e. the contract that maximizes the "Nash product" = the product of the agent's utility functions) is a widely-used approach for identifying the most "fair" contract from those that make up the Pareto front. As we can see in figure 7.1, the Nash Bargaining Solution divides utility equally amongst the negotiating parties, in a linear domain. It can be proven, in fact, that there is a unique Nash bargaining solution for negotiations with convex pareto fronts, which is satisfied trivially for negotiations with linear utilities [103][1].

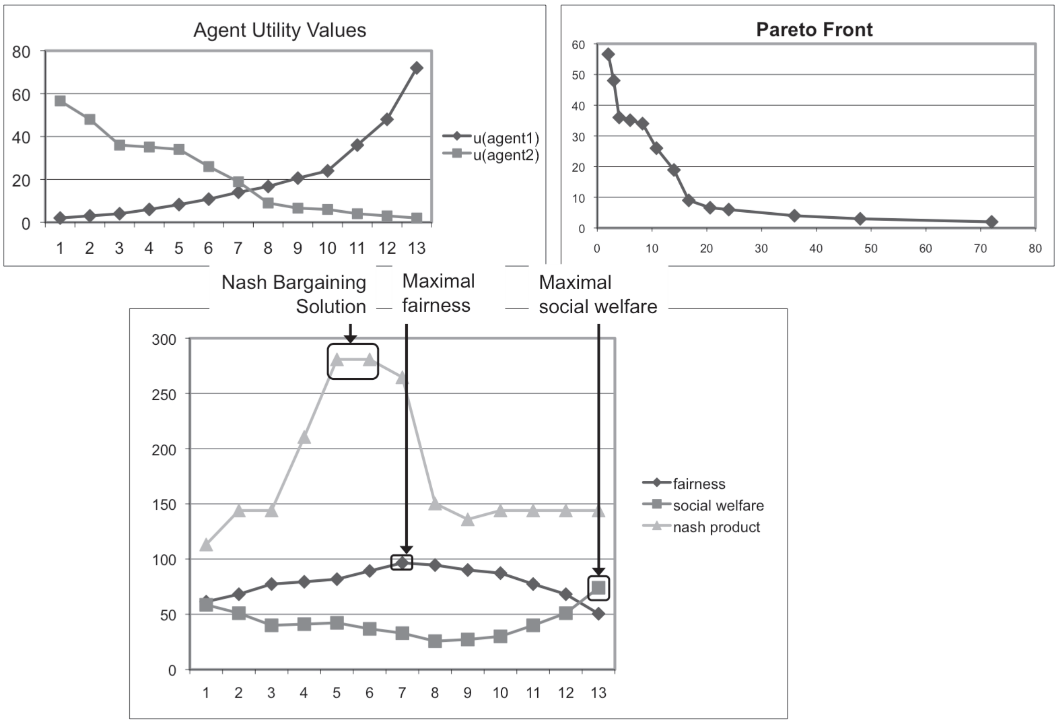

Figure 7.2: The relationship of nash product, fairness and social welfare in a nonlinear utility function

These properties change radically in nonlinear negotiation. As we can see in Figure 7.2, when agents have nonlinear utility functions, the Pareto front can be non-convex [101]. There can be multiple Nash bargaining solutions, even with continuous issue domains, and some of the Nash bargaining solutions may be non-optimal in terms of social welfare and fair division of utility. It is even straightforward to find nonlinear cases where *all* the contracts on the Pareto front are Nash bargaining solutions, despite the fact that many of them diverge widely from maximal fairness and social welfare. The Nash Bargaining Solution concept, which is widely used as a basis for negotiation protocols for linear domains, will

[1]In the discretized issue domains, there can be multiple Nash Bargaining Solutions, but they will all be clustered right next to each other and thus offer similar fairness values.

thus often fare poorly in nonlinear domains. We need, therefore, to find negotiation protocols that can achieve high social welfare and fairness values with nonlinear agent utilities.

7.3 Secure and Fair Mediator Protocol with Approximated Fairness

The Secure and Fair Mediator Protocol (SFMP) was defined to achieve these goals while also protecting agent's private utility information. SFMP consists of two main steps: (1) finding the set of Pareto-optimal contracts, and (2) selecting a fair contract from that set. These steps are defined below.

7.3.1 Finding the Pareto Front

This step is achieved using a mediated approach [36, 37]. One or more mediators propose contracts, initially randomly generated, and ask the agents which ones they prefer. The mediators use this preference information to provide the objective function for a nonlinear optimization technique such as simulated annealing or a genetic algorithm. Over the course of multiple rounds, the mediators converge on the set of pareto-optimal contracts. We assume, as is common in negotiation contexts, that agents prefer not to share their utility functions with others, in order to preserve a competitive edge. Accordingly, our protocol uses a Secure Gathering Protocol based on a Multi-Party Protocol [131] to ensure that mediators can calculate the sum of the agents' utilities without learning, or revealing. the individual agents' utility information. A detailed explanation of the Secure Gathering is given in Appendix.

7.3.2 Selecting the Final Agreement

SFMP selects the final agreement from the Pareto-optimal contract set by calculating which one is the most fair. Several definitions of "fair" have been identified in social choice and game theory [115]. Suppose we have a division

$X = X_1 \cup \cdot \cup X_n$ among n agents where agent i receives X_i. "Simple" fair division results if $u_i(X_i) \geq 1/n$ whenever $1 \leq i \leq n$ (each agent gets at least $1/n$.) Another definition, from game theory, calls a division X is fair if and only if it is Pareto-optimal and envy-free [14]. A division is "envy-free" if no agent feels another has a strictly larger piece of the utility [115].

We adopt simple fair division as our concept of fairness. Contract agreements, in general, rarely fully satisfy this condition. We measure, accordingly, how *close* an agreement is to simple fair division by calculating its "approximated fairness", i.e., the deviation of each agent's utility from the average of the total utility. The approximate fairness of a contract is defined, formally, as follows:

$$V(u_1, \ldots, u_n) = \sum_{i=1}^{n} \frac{(u_i - \overline{u})^2}{n}$$

$(u_1, \ldots, u_n : agent's\ utility\ value\ in\ contract,\ \overline{u} : the\ average\ of\ all\ agent's\ utility\ value)$.

An ideal contract, therefore, has an approximated fairness value of zero, and all other contracts will have larger values. The final agreement selected by our protocol is the contract from the Pareto-optimal set with the smallest approximated fairness value.

Note that our fairness concept is equivalent to the Nash bargaining solution in linear contexts with continuous issue domains. Assume that $u_1 + u_2 + \cdots + u_n = K(constant)$ (where u_i: agent i's utility value). The Nash product is maximized when $u_1 = u_2 = \cdots = u_n = K/n$ (this has been proven mathematically in the field of Isoperimetric Problems). The key difference is that our measure generalizes to nonlinear domains. Approximated fairness does not, however, correspond to the Kalai-Smorodinsky solution because the latter isn't always fair [141].

7.4 Experiments

We ran a series of negotiation simulation experiments in order to demonstrate the weaknesses of the Nash Bargaining Solution in nonlinear domains, and to compare the performance of the SFMP protocol we defined against that of previous approaches. The subsections below describe the experiment setup and results.

7.4.1 Detailed Description of Secure & Fair Mediator Protocol (SFMP)

The SFMP protocol utilizes multiple mediators in order to help assure agent privacy. We assume that there are $k = mn$ mediators M_j and n agents (A_i), where m is an arbitrary integer. Note that this approach requires that m is relatively high if we wish to effectively conceal the agent's private information. If the number of mediators is low, the chances increase that all of the mediators will collude, and thus compromise the agent's privacy.

(Optional Pre-Negotiation Step) Contract space division among mediators: The mediators divide the contract space between them, so each mediator searches a different subregion. Suppose, for example, that there are two issues whose domain is the integers from 0 to 10. In this case, mediator 1 can manage the region of values 0 to 5 for issue 1 and values 0 to 10 for issue 2, while mediator 2 can manage the region of values 6 to 10 for issue 1 and values 0 to 10 for issue 2. This step is optional, but it has the advantage of potentially reducing the time needed to search the contract space by allowing parallel computation.

(Step 1) Secure search to find a Pareto-optimal contract set: Each mediator searches its assigned portion of the contract space using a local search algorithm [123]. We employed Hill Climbing (HC), Simulated Annealing (SA), and Genetic Algorithm (GA) in our experiments. In HC, an agent starts with a random solution and, at each step, makes some random mutations and selects the one that causes the greatest utility increase. When the algorithm cannot find any more improvements, it terminates. In SA, each step of the SA algorithm replaces the current solution by a randomly generated nearby contract, with a probability that depends on the change in utility value and a global parameter T (the virtual temperature) that is gradually decreased during the process. The agent moves almost randomly when the temperature is high, but acts increasingly like a hill climber as the temperature decreases. When T is 0, the search is terminated. The advantage of SA is that it is able to avoid getting stuck in the local optima that occur in nonlinear optimization problems, and often finds more optimal solutions than hill climbing. GA is a search technique inspired by evolutionary biology, using such techniques as inheritance, mutation, selection, and crossover. Initially many individual contracts are randomly generated to form an initial population. After that, at each step, a proportion of the existing population is selected, based on their 'fitness' (i.e. utility values). Crossover and mutation is then applied to these selections to generate the next generation of contracts. This process is repeated until a termination condition has been reached. The objective function for all these local search algorithms is the maximization of social welfare. At each search step, the mediators determine the social welfare values by securely gather the utility values for the current contract(s) from their assigned agents. We call this **secure value gathering**.

(Step 2) Identify agreement: All mediators share the maximum value in their subregion of the contract space with all the other mediators. Based on that, they identify the pareto-optimal contract set. The mediators then select the contract, in that set, that minimizes the approximated fairness metric. This represents the final agreement for that negotiation.

7.4.2 Nash Product Maximization Search (NPMS)

For a comparison case, we used Nash Product Maximization Search (NPMS) to find the Nash bargaining solution for our tests [123]. Our implementation used simulated annealing to maximize the Nash product for the negotiating agents, gathering their utility values using the secure gathering protocol. Simulated annealing has been shown to be very effective for nonlinear optimization tasks [56]. We can use the results of NPMS to assess the scale of the performance decrement caused by using the Nash Bargaining Solution concept in nonlinear domains.

7.4.3 Settings

We conducted five experiments to evaluate the effectiveness of our approach. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions. The number of agents was six, and the number of mediators was four. The mediators could calculate the sum of the agent's utility if four mediators got together. The search space was divided equally amongst the mediators. The domain for the issue values was $[0, 9]$. The constraints included

10 unary constraints, 5 binary constraints, 5 trinary constraints, and so on (a unary constraint relates to one issue, a binary constraint relates to two issues, and so on). The maximum value for a constraint was $100 \times (number\ of\ issues)$. Constraints that satisfy many issues thus have, on average, larger utility, which seems reasonable for many domains. In a meeting scheduling domain, for example, higher order constraints concern more people than lower order constraints, so they are more important. The maximum width for a constraint was 7. The following constraints, for example, are all valid: Issue 1 = $[2, 6]$, Issue 3 = $[2, 9]$.

We compared the following negotiation protocols: SFMP (SA), SFMP (HC), SFMP (GA), Nash Product Maximization Search (NPMS), Basic Bidding protocol, and Exhaustive Search.

(A) SFMP (SA): "SFMP (SA)" is SFMP using Simulated Annealing as the optimization algorithm. The initial temperature was 50 degrees. The initial temperature was 50 degree. For each iteration, the temperature decreased 0.1 degrees, resulting in 500 iterations. *20 + (Number of issues) × 5* searches were conducted, with the initial start point changed randomly for each search.

(B) SFMP (HC): "SFMP (HC)" is SFMP using Hill Climbing as the optimization algorithm. We employed the random restart hill climbing mechanism [123]. *20 + (number of issues) × 5* searches were conducted, with the initial start point changed randomly for each search.

(C) SFMP (GA): "SFMP (GA)" is SFMP using a Genetic Algorithm as the optimization algorithm. The population size was *20 + (number of issues) × 5*. We employed a basic crossover method in which two parent individuals were combined to produce two children (one-point crossover). The fitness function was the sum of all agents' (declared) utility. 500 iterations were conducted. Mutations happened at very small probability. In a mutation, one of the issues in a contract vector was randomly chosen and changed.

(D) Nash Product Maximization Search (NPMS): "Nash Product Maximization Search" used SA to search for the Nash bargaining solution(s), i.e. for contracts that maximize the Nash product. The initial temperature was 50 degrees. For each iteration, the temperature decreased 0.1 degree, resulting in 500 iterations. *20 + (Number of issues) × 5* searches were conducted , with the initial start point changed randomly for each search. These settings are the same as those for SFMP (SA).

(E) Basic Bidding protocol: "Basic Bidding protocol" is the protocol proposed in [56]. In this protocol, the number of samples taken during random sampling is (number of issues) $\times$ 200. The threshold used to remove contract points that have low utility is 200. The limitation on the number of bids per agent is $\sqrt[N]{6,400,000}$ for N agents. This method fails to reach agreements if the number of issues is more than eight because this method has too much computational complexity.

(F) Exhaustive Search: "Exhaustive search" is a centralized brute force algorithm that traverse the entire contract search space to find the Pareto-optimal contract set. The final contract was then selected using our approximated fairness measure. This approach was only computationally practical when the number of issues was seven or fewer.

Our code was implemented in Java 2 (1.5) and run on a core 2-duo processor iMac with 1.0 GB memory on the Mac OS X 10.5 operating system.

7.4.4 Experimental Results

Figure 7.3 compares the social welfare achieved by these six methods. The evaluation measure we used was the *(social welfare for final agreement from method) / (social welfare for final agreement from SFMP (SA))*. As predicted, we found that SFMP (SA) and SFMP (GA) performed better than NPMS, confirming our claim that the Nash Bargaining Solution produces sub-optimal outcomes when applied to nonlinear negotiation. SFMP (SA) and SFMP (GA) had about equal performance. Neither produced fully optimal results, reflecting the difficulty of performing optimization in large nonlinear contract spaces. All the SFMP protocols performed better than the Basic Bidding Protocol, which was hampered by the limit on the number of bids per agent necessitated by the combinatorics of winner determination in this protocol. The performance of SFMP (HC) decreased rapidly as the number of issues grew, because hill climbing got stuck on local optima. The performance of SFMP (SA) and SFMP (GA) did not decrease appreciably as the number of issues increased.

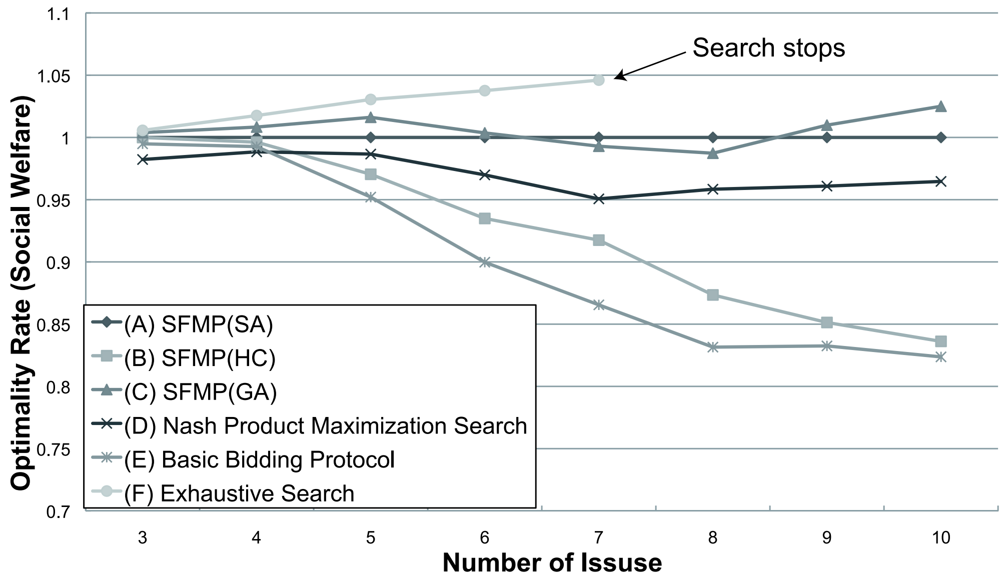

Figure 7.3: Comparison of social welfare

Figure 7.4 compares the number of Pareto-optimal contracts found by the six methods. In this experiment, we limited the domain of each issue to [0,4], so all Pareto-optimal contracts could found, in a reasonable amount of time, using the exhaustive search algorithm. We found that SFMP(SA) and SFMP(GA) were better at finding Pareto-optimal contracts than either the Nash Product Maximization Search or the Basic Bidding Protocol. This makes sense, since the SFMP was explicitly designed to find the entire Pareto front first, before selecting a final agreement, while the other protocols were not. We also found that SFMP(SA) and SFMP(GA) performed better than the Basic Bidding protocol, because the latter often fails to find Pareto-optimal solutions due to the limit on the number of bids allowed by each agent. As always, the performance of SFMP (HC) decreased rapidly as the number of issues grew. SFMP (GA) showed the highest performance on this measure, because GA is inherently more suitable for finding Pareto-optimal contract sets. However, for all methods, when the number of issues increased, the percentage of pareto-optimal contracts found decreased drastically.

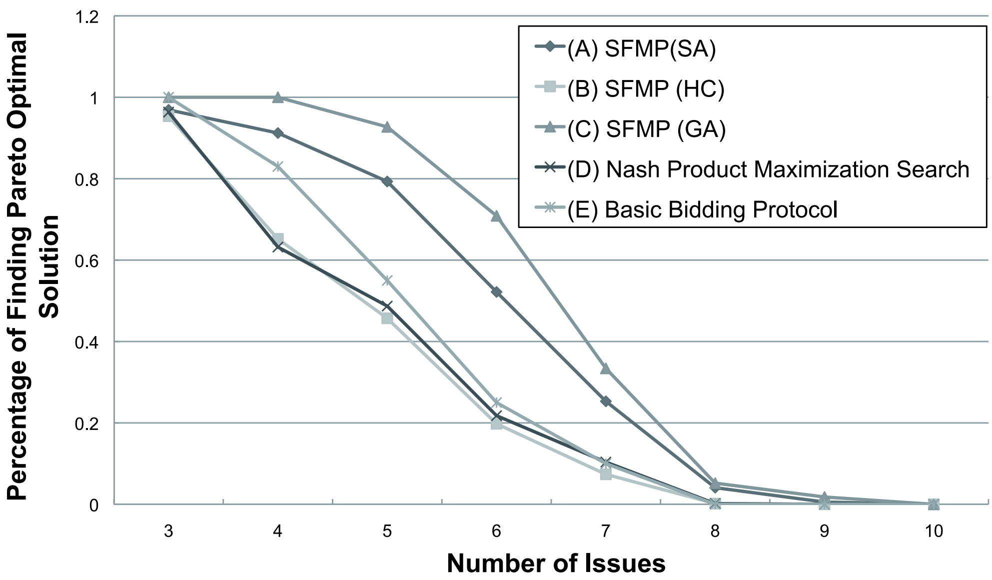

Figure 7.4: Finding Pareto-optimal contracts

Figure 7.5 assessed fairness by comparing the variance of the agent's utilities for the final agreements, across the six methods. Lower variance is better, because it means that utility is distributed more fairly across the agents. The SFMP protocols showed better performance than the Basic Bidding Protocol on this measure because the basic bidding protocol doesn't consider fairness when finding agreements. SFMP (GA) showed the lowest (best) value among the SFMP variants. NPMS outperformed the SFMPs on this measure. This is counter to what we predicted: in nonlinear domains, we would expect the Nash bargaining solutions to vary widely in their fairness values, causing NPMS to produce, on the average, sub-optimal fairness values.

We can potentially explain these results by considering the allocation of computational effort in nonlinear optimization. In an even moderately large nonlinear optimization problem, the contract space is too large to explore exhaustively. If we have only 10 issues with 10 possible values per issue, for example, this produces a space of 10^{10} (10 billion) possible contracts. As a result, with limited computational resources, we have no guarantee of finding the complete Pareto

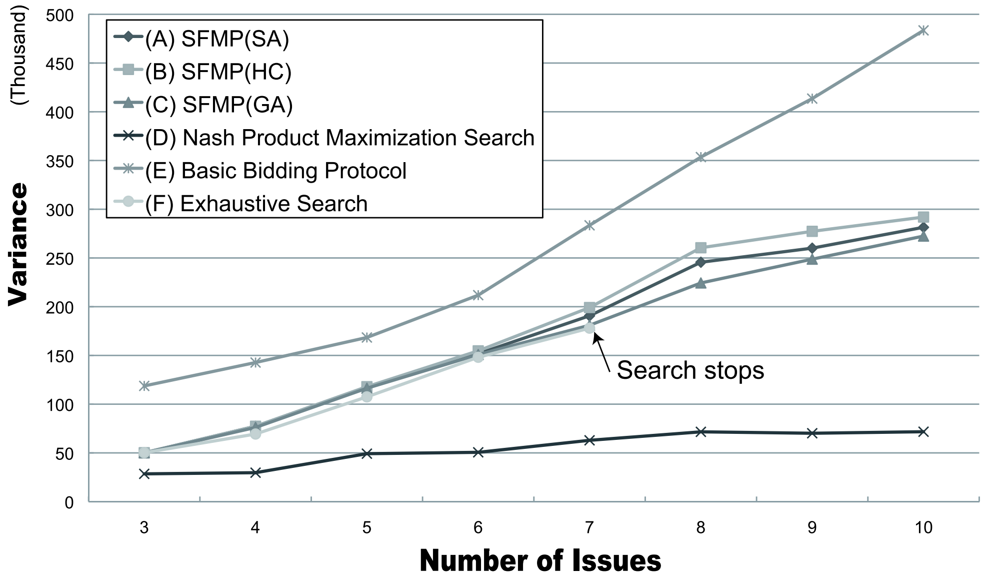

Figure 7.5: Comparison of variance

front. SFMP is presumably only able to find a subset of the Pareto-optimal contracts, and those are scattered over the entire frontier. Because the coverage is sparse, SFMP will often not happen to find the Pareto-optimal contract that optimizes the fairness metric. This will reduce the average fairness score for SFMP. NPMS, by contrast, devotes it's entire computational effort to finding a single Nash-product-maximizing contract. Even though it is an inferior optimization objective, it has the benefit of a more concentrated application of computing resources.

This interpretation is supported by figure 7.6, which shows the utility values for SFMP and NPMS for a case with two agents and five issues, with randomly generated nonlinear utility functions. The diamond-shaped points show the contracts considered by NPMS, while the box-shaped points show the contracts considered by SFMP. Since SFMP aims to find the entire Pareto front, it searches throughout the Pareto frontier. NPMS, by contrast, aims to find the contract that directly maximizes the Nash product, so it focuses it's search toward the middle of the Pareto frontier. As figure 7.6 shows, SFMP in this case got closer

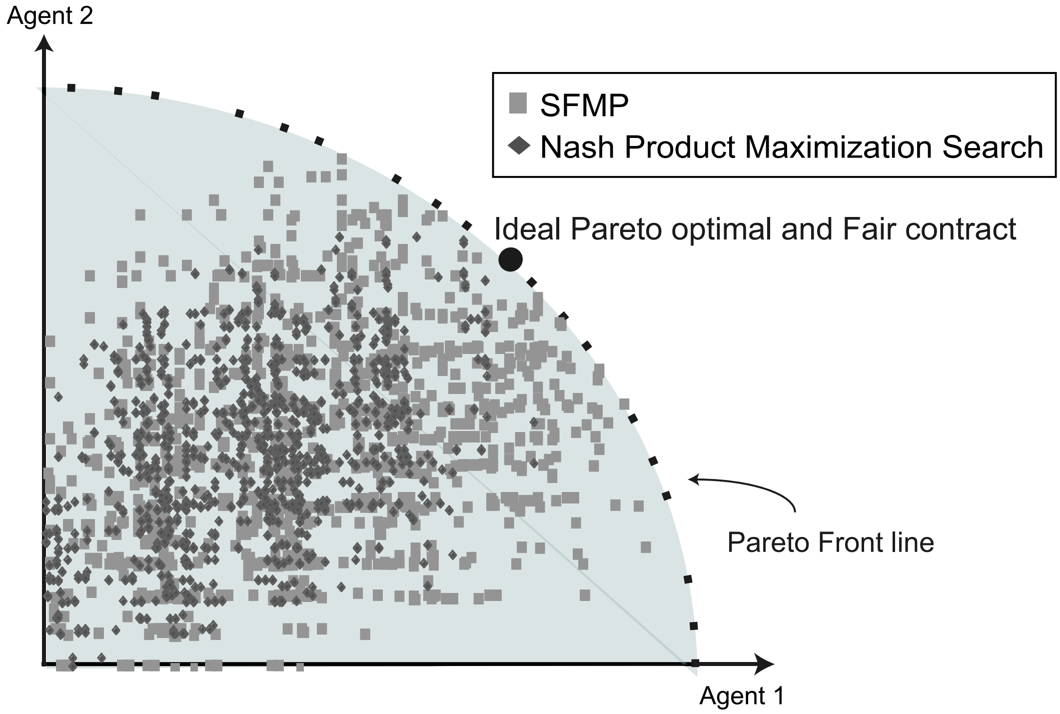

Figure 7.6: Comparison between SFMP and Nash Product Maximization Search

to the Pareto frontier than NPMS.

Figure 7.7 compares the failure rates across the six methods, to assess their scalability of our methods. For all the methods, if the computing time method exceeded 100 seconds, the negotiation was aborted and it was treated as a failure. The failure rate for the Basic Bidding and exhaustive search protocols increased exponentially with the number of issues. This is because that the computational complexity of finding agreements in these protocols is quite large. All the other protocols had negligible failure rates.

The key experimental results can be summarized as follows:

- SFMP, as predicted, maximizes social welfare more effectively than NPMS. It also out-performs the Basic Bidding protocol.

- SFMP finds fairer contracts than the Basic Bidding Protocol, but is less

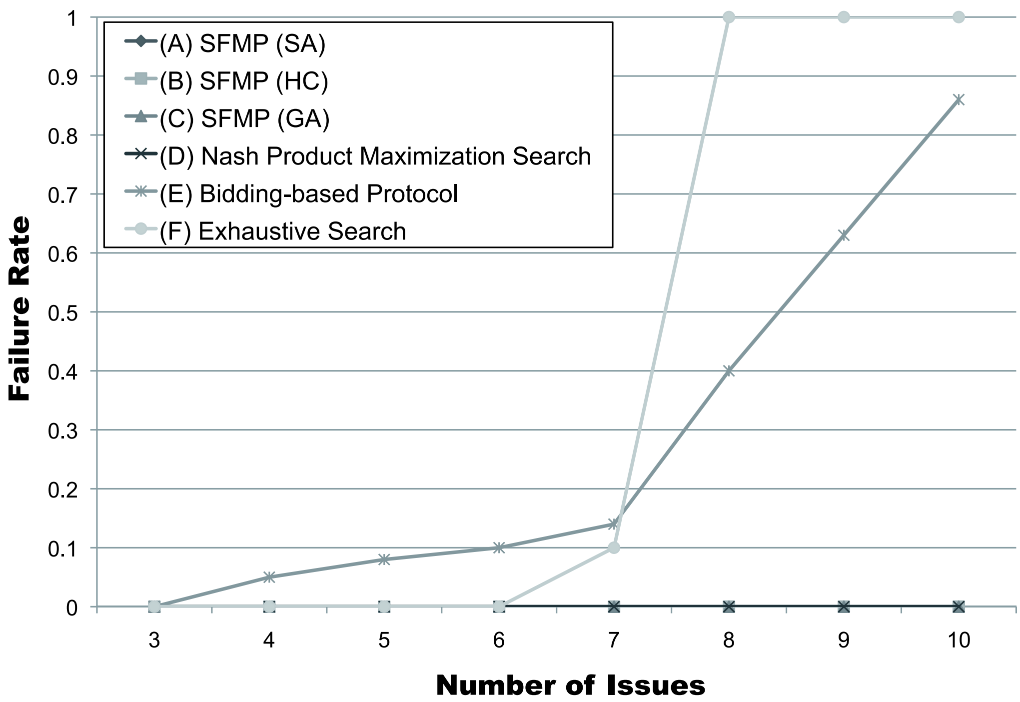

Figure 7.7: Comparison of Failure Rate

fair than NPMS.

- SFMP has a lower failure rate (and thus greater scalability) than the Basic Bidding Protocol.

We also found that the negotiation methods were sensitive to the complexity of negotiation setting, due to the combinatorics of the local search algorithms they employed. The larger the number of issues, the lower the optimality of the outcomes.

7.5 Conclusion

We showed that the Nash Bargaining Solution, although provably optimal for negotiations with linear utilities, can lead to sub-optimal outcomes when applied to nonlinear negotiations. We also presented the Secure and Fair Mediator Protocol (SFMP), a novel negotiation protocol that utilizes a combination of nonlinear optimization, secure information sharing, and an approximated fairness metric, and demonstrated that it achieves higher social welfare values than a protocol based on searching for the Nash bargaining solution. Finally, we demonstrated that SFMP out-performs our own previous efforts to enable multi-lateral negotiations in complex domains.

Appendix: Secure value gathering

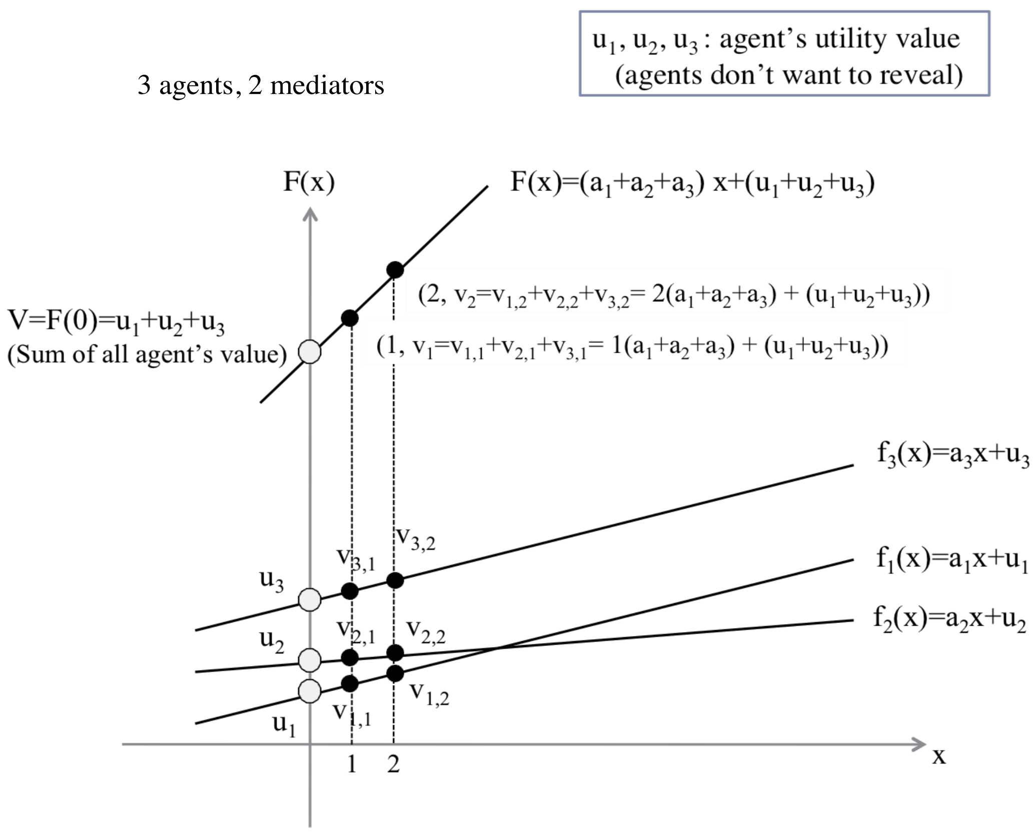

Figure 7.8: The example of Secure value gathering

Appendix 1 below includes an explanation of secure value gathering. Figure

7.8 shows an example with three agents and two mediators ($k = 2$). This is just for illustrative purposes: in practical situations, k should be larger to reduce the likelihood of mediator collusion. In the following, u_i is agent i's utility value.

1. The mediators ask the agents to generate "shares" v. Each agent A_i will send one share $v_{i,j}$ to each mediator M_j.

2. Each agent i (A_i) randomly selects a k dimensional polynomial formula which fulfills $f_i(0) = u_i$, In figure 7.8, for example, agent 1 selected $f_1(x) = a_1x + u_1$, agent 2 selected $f_2(x) = a_2x + u_2$ and agent 3 selected $f_3(x) = a_3x + u_3$.

3. Each agent (A_i) calculates a share $v_{i,j} = f_i(j)$ for each mediator (M_j) and sends it to that mediator. For example, agent A_1's share for mediator M_2 would be $v_{1,2} = f_1(2) = 2a_1 + u_1$.

4. Every mediator j (M_j) sums the shares $v_{1,j}, \ldots, v_{n,j}$ it receives from the agents in order to calculate $v_j = v_{1,j} + \cdots + v_{n,j}$. In Figure 7.8, for example, mediator 1 received the shares $v_{1,1},\ v_{2,1},\ and\ v_{3,1}$ and calculated $v_1 = v_{1,1} + v_{2,1} + v_{3,1}$.

5. The j mediators add their share sums v_j together to calculate $F(x)$ for x from 1 to j. Using Lagrange's interpolating polynomial, it is then straightforward to calculate $F(0)$, which corresponds to the sum of all the agent's utility values for a contract. The net result is that the social welfare is calculated without any one mediator knowing the utility of any contract for any individual agent.

8

An Approach to Scalable Multi-issue Negotiation: Decomposing the Contract Space based on Issue Interdependencies

8.1 Introduction

In this chapter, we propose a new protocol in which a mediator tries to reorganize a highly complex utility space into several tractable utility subspaces, in order to reduce the computational cost. Issue groupings are generated by a mediator based on an examination of the issue interdependencies. First, we have to define a measure for the degree of interdependency between issues. We define four such measures. Second, we generate a weighted non-directed interdependency graph 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 [140], this previous work doesn't identify optimal issue groups. Finally, we demonstrate that our protocol, based on issue-groups, has higher scalability than previous efforts, and discuss the impact on the optimality of the negotiation outcomes.

The remainder of this chapter is organized as follows. In 8.2, we describe several measures for assessing the degree of issue interdependency. In 8.3, we

ID	Issue1	Issue2	Issue3	Issue4	Utility
1	$[2,4]$	$\emptyset$	[4,6]	$\emptyset$	20
2	$\emptyset$	5	[3,7]	[1,6]	40
3	[3,8]	$\emptyset$	$\emptyset$	$\emptyset$	25
4	4	[2,7]	9	[4,5]	50

Table 8.1: Utility function for an agent

present a technique for finding issue sub-groups, and propose a protocol that uses this information to enable more scalable negotiations. In 8.4, we present the experimental results. Finally, we draw conclusions.

8.2 Interdependency Rate and Interdependency Graph

A issue interdependency for multi-issue negotiations is defined as follows: If there is a constraint between i_X and i_Y, then we assume i_X and i_Y are interdependent. If, for example, an agent has a binary constraint between issue 1 and issue 3, issue 1 and issue 3 are interdependent for that agent - see Table 8.1.

The strength of issue interdependency is measured by *interdependency rate*. We define four measures for the interdependency between *issue* i_j and *issue* i_{jj} for agent a:

(A) Number of constraints only: $D_a^{(A)}(i_j, i_{jj}) = \sharp\{c_k | \delta_a(c_k, i_j) \neq \emptyset \wedge \delta_a(c_k, i_{jj}) \neq \emptyset\}$. This measures the number of constraints that inter-relate the two issues.

(B) Number of terms of constraints: $D_a^{(B)}(i_j, i_{jj}) = \sum_{c_k \in C} \epsilon_a(c_k)\ if\ c_k\ is$ $\delta_a(c_k, i_j) \neq \emptyset \wedge \delta_a(c_k, i_{jj}) \neq \emptyset$. This sums the order of the constraints relating two issues, based on the intuition that higher-order constraints are more important than lower-order (e.g. binary) constraints.

(C) Utility value of constraints: $D_a^{(C)}(i_j, i_{jj}) = \sum_{c_k \in C} v_a(c_k)\ if\ c_k\ is\ \delta_a(c_k, i_j) \neq \emptyset \wedge \delta_a(c_k, i_{jj}) \neq \emptyset$. This sums the weights of the constraints that inter-relate the two issues.

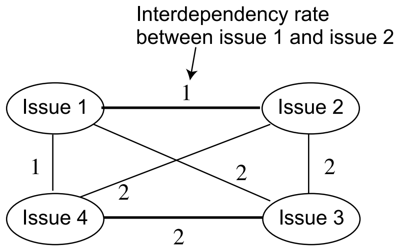

Figure 8.1: Interdependency Graph

(D) Number of terms and utility of constraints: $D_a^{(D)}(i_j, i_{jj}) = D_a^{(B)}(i_j, i_{jj}) * D_a^{(C)}(i_j, i_{jj})$. This is the product of measures B and C. In addition, we assume that $D_a^{(B)}(i_j, i_{jj})$ and $D_a^{(C)}(i_j, i_{jj})$ are normalized.

The agents capture issue interdependency information as an interdependency graph. An interdependency graph is represented as a weighted non-directed graph, in which a node represents an issue, an edge represents the interdependency between issues, and the weight of an edge represents the interdependency rate between the issues. An interdependency graph is thus formally defined as:

$$G(P, E, w) : P = \{1, 2, \ldots, |I|\}(finite\ set),$$
$$E \subset \{\{x, y\}|x, y \in P\}, w : E \rightarrow R.$$

Figure 8.1 shows the interdependency graph for the constraints listed in Table 8.1.

8.3 Negotiation Protocol based on issue interdependency

Our proposed negotiation protocol works as follows. A mediator gathers private issue interdependency graphs from each agent, generates a social interdependency graph, identifies issue sub-groups, and then uses that information to guide the search for a final agreement. In fact, we apply the concept of issue-grouping to the Basic Bidding in our negotiation protocol. In Basic Bidding Protocol, agents can make agreement without submitting all agents' privacy information, however, the scalability is not so high. By applying the concept of grouping-issues to the Basic Bidding Protocol, we can propose high scalable protocol considering the agents' privacy. We describe the details below:

[**Step 1: Analyzing issue interdependency**] Each agent analyzes issue interdependency in its own utility space, using Algorithm 1, and generates an interdependency graph. Each agent sends its' interdependency graph to the mediator.

Algorithm 5 get_Interdependency(C)

C: a set of constraints

```
for c ∈ C do
  for i := 0 to Number of issues do
    for j := i + 1 to Number of issues do
      if Issue i and Issue j are interdependent in c then
        Calculate interdependencyGraph[i][j]
      end if
    end for
  end for
end for
```

[**Step 2: Grouping issues**] In this step, the mediator identifies the issue-groups. First, the mediator generates a social interdependency graph from the private interdependency graphs submitted by the agents. A social interdependency graph is almost same as a private interdependency graph. The only difference is that the weight of an edge represents the *social* interdependency rate. The social interdependency rate between $issue\ i_j$ and $issue\ i_{jj}$ is defined as: $\sum_{a \in N} D_a(i_j, i_{jj})$. ($D_a(i_j, i_{jj})$: Interdependency rate between $issue\ i_j$ and $issue\ i_{jj}$ by agent a).

Next, the mediator identifies the issue-groups based on the social interdependency graph. In this protocol, the mediator tries to find optimal issue-grouping using simulated annealing (SA) [123]. The evaluation function for the simulated annealing is the sum of the weights of the edges that do not span separate issue-groups. The goal is to maximize this value. Figure 8.2 shows an example of evaluation values for two issue-groups. In Figure 8.2 (A), the evaluation value is 8 because there are non-spanning edges between issue 1 and issue 2, issue 3 and issue 4, issue 3 and issue 5, and issue 4 and issue 5. In Figure 8.2 (B), the evaluation value is 9 because there are non-spanning edges among issue 1, issue

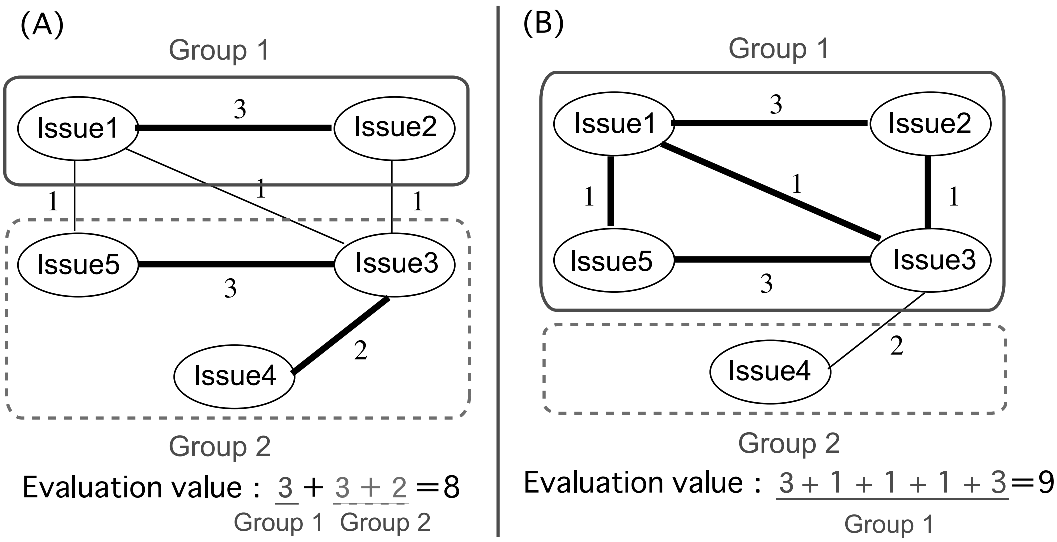

Figure 8.2: Evaluation value in identifying issue-groups

2, issue 3, and issue 5. The number of issue-groups is decided before the protocol begins.

Agents are at risk for making an agreement that is not optimal for themselves by dividing the interdependent issues. In other words, there is the possibility of making a low utility agreement by ignoring the interdependency of some issues. However, agents can make a better agreement in this protocol because the mediator identifies the issue-groups based on the rate of interdependency.

[**Step 3: Generating bids**] First, each agent generates bids for the entire set

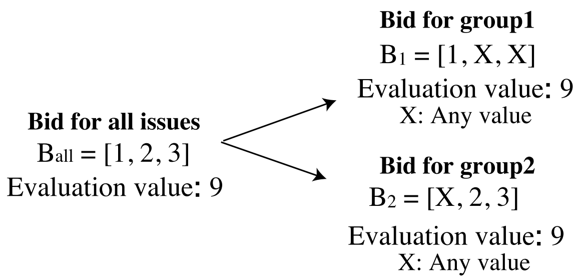

Figure 8.3: Division for the bid by agents

of issues using the bidding-based protocol [56]. Concretely speaking, each agent samples its entire utility space in order to find high-utility contract regions. After that, each agent uses a nonlinear optimizer based on simulated annealing [123] to try to find the local optimum in its neighborhood. 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(threshold), then the agent defines a bid that covers all the contracts in the region that have that utility value.

Next, agents divide these bids into sub-bids for each issue-group, and determine their valuations for each sub-bid. Agents set their valuation for a bid to be the utility of the highest-value contract in the bid region. In Figure 8.3, for example, an agent selects the global bid $B_{all} = [1, 2, 3]$ for all issues, and divides B_{all} into sub-bids $B_1 = [1, X, X]$ for issue group 1 and $B_2 = [X, 2, 3]$ for issue group 2 (X: any value). In this case, the agent's evaluations for both sub-bids are 9.

[Step 4: Finding the Solutions] 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 social welfare (i.e. the highest summed bid value). The mediator employs breadth-first search with branch cutting to find the social-welfare-maximizing bid overlaps. After that, the mediator finds the final contract by consolidating the winning sub-contracts from each issue-group.

In terms of an agent's strategic behavior, we assume agents are truthful. In addition, theoretically, our protocol can be made incentive-compatible (i.e. where agents are given incentive to provide the truthful bid values that are necessary to ensure [near-]optimal social welfare) if we employ the Groves mechanism [47] with some theoretical assumptions on unlimited budgets and unlimited computational resources. Also, we must assume that the cost (payment) does not depend on the

[1] A bid can specify not just a specific contract but an entire region. For example, if a bid covers the region [0,2] for issue 1 and [3,5] for issue 2, the bid is satisfied by the contract where issue 1 has value 1 and issue 2 has value 4. For a combination of bids to be consistent, the bids must all overlap.

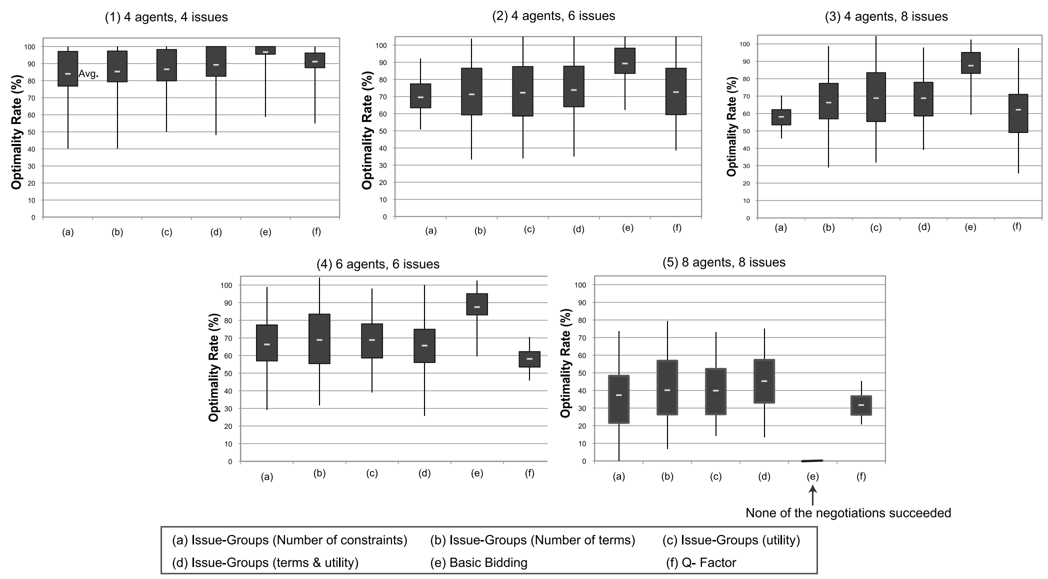

Figure 8.4: Box-plots of the optimality rate

other issues. Then, we can define agent i's utility function as follows: $u_i = v_i - c_i$, where v_i is value of agreement when some multiple issues are satisfied and c_i is the payment computed by one of the Grove's mechanisms. We describe the details in the appendix.

8.4 Experimental Results

8.4.1 Setting

We conducted several experiments to evaluate our approach. In each experiment, we ran 100 negotiations. The following parameters were used. The domain for the issue values was $[0, 9]$. The number of constraints was 10 unary constraints, 5 binary constraints, 5 trinary constraints, and so on. (a unary constraint relates to one issue, a binary constraint relates to two issues, etc). The maximum value for a constraint was *100* $\times$ *(Number of Issues)*. Constraints that satisfy many issues have, on average, larger utility, which seems reasonable for many domains. In the meeting scheduling domain, for example, higher order constraints concern more people than lower order constraints, so they are more important. The maximum width for a constraint was 7. The following constraints would all be valid: Issue $1 = [2, 6]$, Issue $3 = [2, 9]$.

We compare the following six methods: “(a) Issue-groups (Number of constraints),” “(b) Issue-groups (Number of terms),” “(c) Issue-groups (utility),” “(d) Issue-groups (terms & utility),” “(e) Basic Bidding,” and “(f) Q-Factor.” (a)-(d) are variants of the issue-group protocol proposed in this chapter, using the four different interdependency rate measures $D_n^{(A)} \sim D_n^{(D)}$ we described above. This allows us to compare the efficacy of the different interdependency rate measures. “(e) Basic Bidding” is the bidding-based protocol proposed in [56], which does not employ issue-grouping. In this protocol, agents generate bids by finding the highest utility regions in their utility functions, and the mediator finds the optimum combination of bids submitted from agents. “(f) Q-Factor” is the Maximum Weight Interdependent Set (MWIS) protocol proposed in [97, 98]. MWIS is a variant of bidding protocol where agents use the Q-factor, a combination of region and utility, to decide which bids to submit. This reduces the failure rate because agents are less likely to submit low-volume bids that do not overlap across agents.

The parameters for generating bids in (a)-(f) are as follows [56]. The number of samples taken during random sampling is *(Number of Issues)* $\times$ *200*. The starting temperature for the simulated annealing algorithm used to find high points near the samples is 30 degrees . For each iteration, the temperature decreases 1 degree, so the annealer runs for 30 iterations. Note that it is important that the annealer does not run too long or too hot because then each search will tend to find the global optimum instead of the peak of the optimum nearest the sampling point. The threshold used to cut out contract points that have low utility is 100. The limitation on the number of bids per agent is $\sqrt[N]{6,400,000}$ for N agents, because it was only practical to run the deal identification algorithm if it explored no more than about 6,400,000 bid combinations. The parameters for identifying issue sub-groups, in (a)-(d), are as follows. The initial temperature for the simulated annealing algorithm is 30 degrees. For each iteration, the temperature decreased 3 degrees, producing a total of 10 iterations. The number of issue-groups generated is three. In "(f) Q-Factor," Q (Q-Factor) is defined as $Q = u^{\alpha} * v^{\beta}$($u$: utility value, v: volume of the bid or constraint), $\alpha = 0.5$, $\beta = 0.5$.

We used simulated annealing (SA) [123] to approximate the optimum social welfare for each negotiation test run. Exhaustive search was not a viable option because it becomes computationally intractable as the number of issues grows. The SA initial temperature is 50.0 and decreases linearly to 0 over the course of 2,500 iterations. The initial contract for each SA run is randomly selected. The optimality value for a negotiation run, in our experiments, is defined as *(The social welfare achieved by each protocol) / (The social welfare calculated by SA)*.

Our code is implemented in Java 2 (1.5) and run on a core 2-duo CPU with 1.0 GB memory on a Mac OS X (10.6).

8.4.2 Experimental Results

Figure 8.4 compares the optimality rate of the different protocols. The lines represent the min and max values, the boxes represent *+/- 1* standard deviation, and the '-' represents the average. The optimality rate of our method ((a)-(d)) is higher than "(f) Q-Factor" when the number of issues is large. In addition, "(d) Issue-Groups (terms & utility)" produces a higher optimality rate than (a). In *t-test*, there is a significant difference between (a) and (d) in case 5 ($t(198) = 0.003$, $P < 0.05$, one-sided testing). Therefore, the interdependency rate measure based on constraint utility and number of constraint terms works best of those we tried. "(e) Basic Bidding" produces the highest optimality scores in case 1 and case 2 where it does not fail. However, (e) succeeded for none of the negotiations in case 3, so it's scalability is limited.

Figure 8.5 compares the failure rates. The failure rate of our method ((a)-(d)) is lower than "(e) Basic Bidding", especially as the number of issues increases. Also, our method ((a)-(d)) has essentially the same (very low) failure rate as "(f) Q-Factor." Our proposed method and Q-Factor thus achieve the same reduction in failure rate by different means: one by negotiating by issue-groups, the other by Basic Bidding Protocol on the quality factor. It should be noted, however, that the Q-factor approach is probably *not* incentive-compatible. While using the Q-factor to pick bids does reduce the failure rate, there is an incentive for agents to

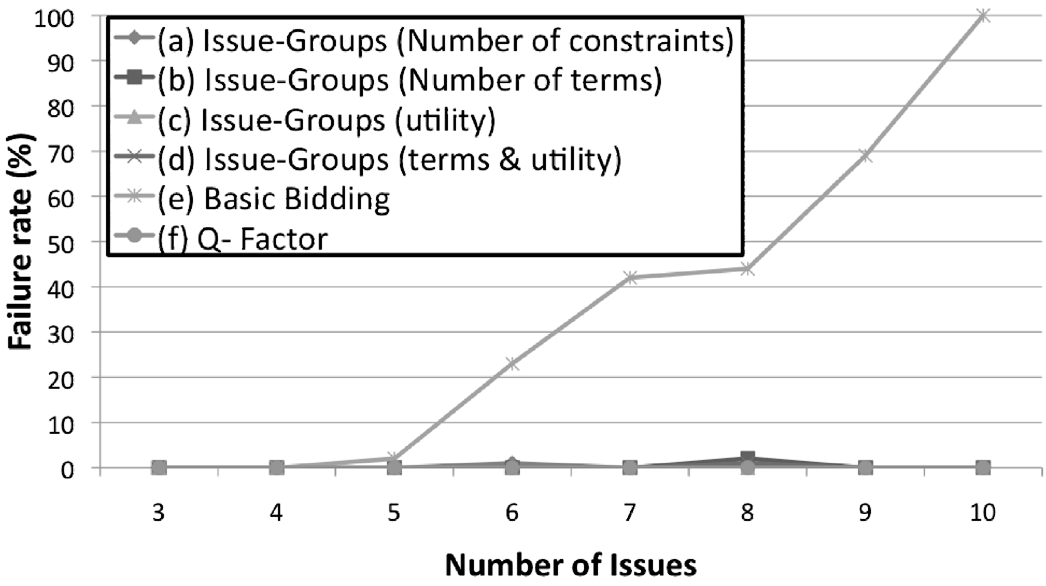

Figure 8.5: Failure rate

cheat and submit bids based only on their utility. This increases the likely utility of the final deal, for them, and may not substantially increase the probability of a failed negotiation *if the other agents do not cheat as well*. This thus creates an prisoner's dilemma game, such that all agents are individually incented to take actions that make things worse for everybody. Our issue-clumping protocol, by contrast, does not require that agents selflessly prefer higher volume bids, and thus avoids this incentive compatibility problem.

Figure 8.6 shows the optimality rate and failure rate as a function of the number of issue subgroups in our protocol, for experiments with four agents. The optimality rate decreases as the number of issue subgroups increases. This is because the possibility that important interdependencies cut across issue subgroups (and are thus ignored) increases when there are more subgroups. On the other hand, the failure rate for making agreements decreases as the number of issue subgroups increases. This is because the number of issues in each issue subgroup decreases, and the computational cost for finding agreements becomes smaller, thereby reducing the likelihood of missing an agreement and therefore having a

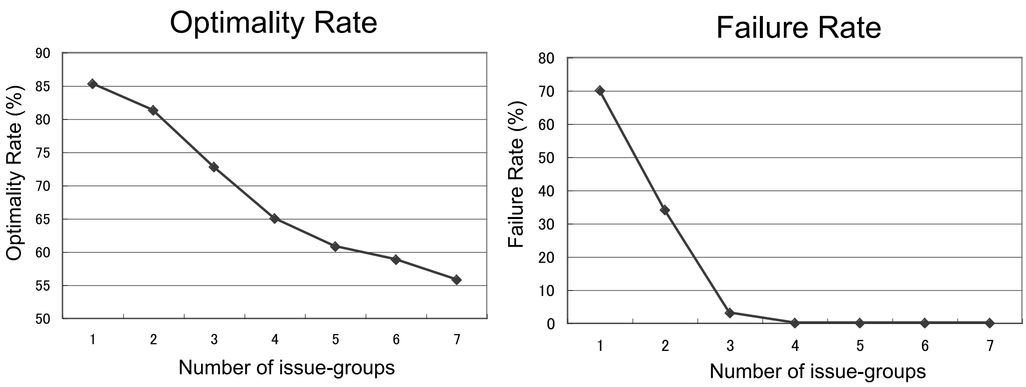

Figure 8.6: Effect of the number of issue-groups

failed negotiation. Thus, there is a trade-off between the optimality rate and the failure rate in selecting the number of issue groups in our protocol.

8.5 Conclusion

In this chapter, we proposed a new negotiation protocol, based on grouping issues, which can find high-quality agreements in interdependent issue negotiation. In this protocol, agents generate their private issue interdependency graphs, the mediator identifies the issue-groups based on these graphs, and multiple independent negotiations proceed for each issue sub-group. We demonstrated that our proposed protocol has greater scalability than previous work, and analyzed the effectiveness of different measures of the interdependency rate. For future work, we will investigate how to improve optimality while maintaining the failure rate advantages of our protocol. One possible track, for example, is to select the number of issue groups adaptively based on the issue dependency topology. Another is to conduct additional negotiation, after the concurrent sub-contract negotiations, to try to increase the satisfaction of constraints that crossed sub-contract boundaries.

Appendix: Incentive Compatibility

Our negotiation protocol can be made incentive compatible by defining payments for agents and employing Groves mechanism[47]. We assume unlimited agent budgets, which is a standard assumption for these kinds of incentive analyses [5]. We also assume each agent knows its own utility space completely and can find the optimal points without any cost. We call the new mechanism (protocol) $\mathcal{M}$. We define agent is type θ_i to be a set of constraints C_i and its value w_i: $\theta_i = (C_i, w_i)$, where $w_i = \sum_{c \in C_i} w(c)$. θ_i can be viewed as a bid from agent i. In this mechanism, agent i submits type $\hat{\theta}(a\ bid)$, which may not be true (i.e. may not represent the true weight for those constraints). Based on the reported types $\theta = (\theta_1, \ldots, \theta_N)$, our mechanism computes: $s^*(\hat{\theta}) = arg_{s \in S, s\ is\ consistent} max \sum_i z_i(s, \hat{\theta}_i)$, where S is a set of contracts, $z_i(s, \hat{\theta}_i)$ is agent i's valuation function on the consistent contract s when i reports $\hat{\theta}_i$. s does not violate any constraints in $\hat{\theta}$. $z_i(s, \hat{\theta}_i))$ is a nonlinear function in our case. For the purpose of this analysis, we will assume an ideal case in which each agent has complete knowledge on his/her own utility space. We define agent is payments as follows a direct adaptation of Groves mechanism: $t_i(\hat{\theta}) = h_i(\hat{\theta}_{-i}) - \sum_{j \neq i} z_j(s^*(\hat{\theta}), \hat{\theta}_j)$ – (1) The first term, $h_i(\hat{\theta}_{-i})$, in the right hand in the equation (2) is an arbitrary function on the reported types of every agent except i. Agent i's utility for making a bid (i.e. reporting a type) $\hat{\theta}_i$ can be defined as follows: $u_i^{\mathcal{M}}(\hat{\theta}_i) = z_i(s^*(\hat{\theta}), \theta_i) - t_i(\hat{\theta})$ – (2)

Proposition 1 (Incentive compatibility). $\mathcal{M}$ is incentive compatible (i.e. truth telling is a dominant strategy).

Proof. The proof is almost the same as that for Grove's mechanism. Based on the utility function (2), $u_i^{\mathcal{M}}(\hat{\theta}_i) = z_i(s^*(\hat{\theta}), \theta_i) - t_i(\hat{\theta}) = z_i(s^*(\hat{\theta}_i), \theta_i) + \sum_{j \neq i}(s^*(\hat{\theta}), \hat{\theta}_j) - h_i(\hat{\theta}_{-i})$. Agent i can not control $h_i(\hat{\theta}_{-i})$. Therefore he wants to maximize $z_i(s^*(\hat{\theta}_i), \theta_i) + \sum_{j \neq i}(s^*(\hat{\theta}), \hat{\theta}_j)(*)$. On the other hand, mechanism $\mathcal{M}$ computes the following because to maximize social welfare efficiency: $argmax_{s \in S} z_i(s, \hat{\theta}_i)$. This can be written as follows: $argmax_{s \in S}[z_i(s, \hat{\theta}_i) + \sum_{j \neq i} z_j(s, \hat{\theta}_j)]$. For agent i, to maximize the equation (*), he must report $\hat{\theta}_i = \theta_i$, i.e. his truthful type.

9

Common Testbed Generating Tool based on XML for Multiple Interdependent Issues Negotiations

9.1 Introduction

Most negotiation protocols are evaluated based on one's own testbed. For example, [56] and [90] are only evaluated on randomly generated utility spaces. However, the effectiveness of the negotiation protocols is evaluated based on the same testbed. Thus, we propose a tool that generates testbeds for evaluating multi-issue negotiation protocols by focusing on the utility function based on cube-based constraints[56] and cone-constraints. Cone-constraints capture the intuition that agent utilities for a contract usually decrease gradually (rather than step-wise) by the distance from their ideal contract, which is described in chapter 6.

We propose a common testbed generating tool based on XML. The input is the configuration files that define the number of issues, the number of agents, etc. The testbed generating 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, A

Single Hill, Two-Hills, and Several Hills. These types of utility spaces are based on actual negotiation settings.

In this chapter, we define XML formats, 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. In this chapter, we show cube-based and cone-based constraint formats that define the building blocks of utility function spaces.

We also demonstrate some examples that use our testbed. We show a JAVA program that searches for agreement contracts in agent utility spaces using Simulated Annealing (SA). In this program, the XML structure is analyzed using Document Object Model (DOM)[144], and then agreement points are searched for.

The remainder of the chapter is organized as follows. In 9.2, we describe a model of nonlinear multi-issue negotiation and propose a testbed generating tool based on XML for multi interdependent issues. In 9.3, we demonstrate examples using our testbed. Finally, we draw a conclusion.

9.2 Common Testbed based on XML for Negotiation Protocols

9.2.1 Testbed Generating Tool

We have been implementing a common testbed generating tool for multi-issue negotiation protocols based on XML. The input of a testbed generating tool is a configuration file that includes the number of issues and the number of agents. The output is an XML file that defines the agents' utility spaces.

Figure 9.1 shows the program flow of our testbed generating tool. First, the utility space is defined based on the configuration file. Second, constraints are generated based on the specified type of utility spaces. Finally, an XML file is outputted. The details of the testbed generating tool are shown as follows:

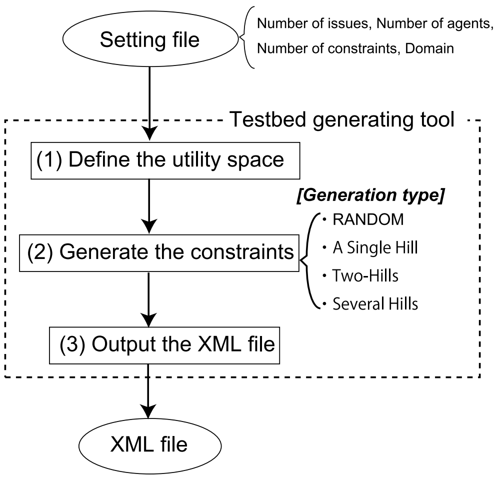

Figure 9.1: Flow of testbed generating tool

(1) Defining utility space: The testbed generating tool defines the utility space information based on the configuration file. The configuration file includes the number of issues, agents, and constraints as well as the value domain per issue. Constraints are classified by the number of related constraints. For example, a unary constraint is related to one issue, a binary constraint is related to two issues, etc. In the configuration file, we write the number of constraints for each related constraint like "unary constraints include 10, binary constraints include 5, etc."

(2) Generating utility spaces: In the current implementation, the testbed generating tool generates utility spaces based on four different types of utility spaces: Random, A Single Hill, Two Hills, and Several Hills. Statements about the details of each type are shown as follows:

Random: In this type, constraints are generated randomly. Such generation is used in the experiments in several works [56]. Figure 9.2 shows an example of utility space plotted by all statements as agent constraints. This utility space plotted is highly nonlinear, as Figure 9.2(A) shows.

A Single Hill: An example of this type is a collaborative negotiation among the same type of agents. The utility space plotted by all agents has

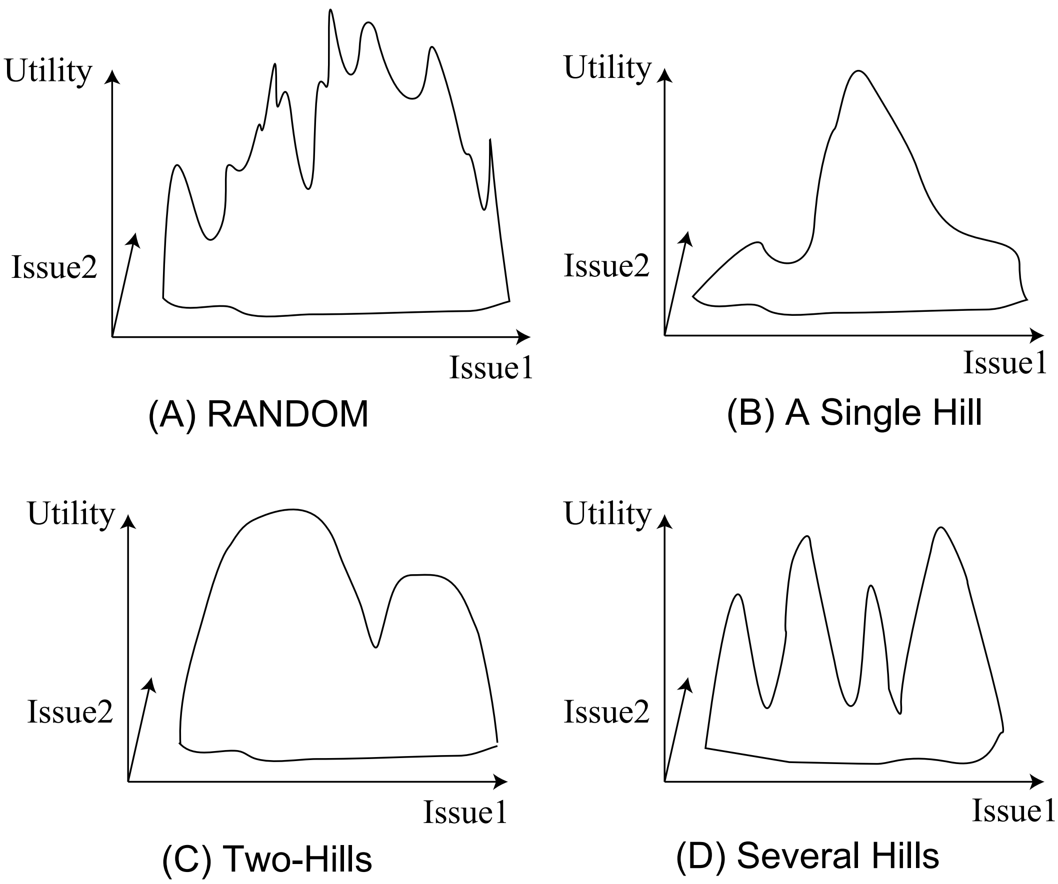

Figure 9.2: Generation type

one higher point, as Figure 9.2(B) shows. In such utility spaces, reaching an agreement is usually easy.

Two Hills: An example of this type is a bilateral negotiation between two types of agents. In particular, such negotiation between buyers and sellers is popular. The utility space plotted by all agents has two higher points, as Figure 9.2(C) shows. In such utility spaces, making agreements is hard because the agents are likely in a hostile relation.

Several Hills: An example of this type is collaborative negotiation among more than three other types of agents. Collaborative design for a car among designers, engineers, and business managers is a concrete example. The utility space plotted by all agents' constraints has more than three higher points, as Figure 9.2(C) shows. In such utility spaces, finding agreement points is hard because there are too many hills. Thus search algorithms usually try to find the highest points.

(3) Output XML file: The testbed generating tool outputs the XML file on the testbed for negotiation. By outputting these files, users can easily understand the information. Additionally, users can modify, change, and update the data, and XML data are not dependent on a certain environment. Users like research communities can also easily exchange data with each other. The details of the XML formats are described in the next subsection.

9.2.2 XML format for testbeds

We propose the XML format for expressing the agent's utility function. In XML, this information is defined by tags. The specification of XML formats in cube-constraints and cone-constraints is described as follows:

```
<?xml version="1.0" encoding="Shift_JIS" standalone="no"?>
<UtilitySpace>
<Dimension>4</Dimension>
<Domain>0-9</Domain>
<Agent no=0 name="Alice">
<ReservationValue>11</ReservationValue>
<Constraint no=0 name="0">
<Cardinality>2</Cardinality>
<Utility>69</Utility>
<Minimum>
<Issue no=2 name="size"> 4 </Issue>
</Minimum>
<Maximum>
<Issue no=2 name="size"> 8 </Issue>
</Maximum>
</Constraint>
<Constraint no=1 name="1">
...
</Constraint>
</Agent>
</UtilitySpace>
<Agent no=1 name="Bob">
<ReservationValue>15</ReservationValue>
<Constraint no=0 name="0">
<Cardinality>1</Cardinality>
...
</Constraint>
</Agent>
</UtilitySpace>
```

Figure 9.3: Example XML for cube-constraints

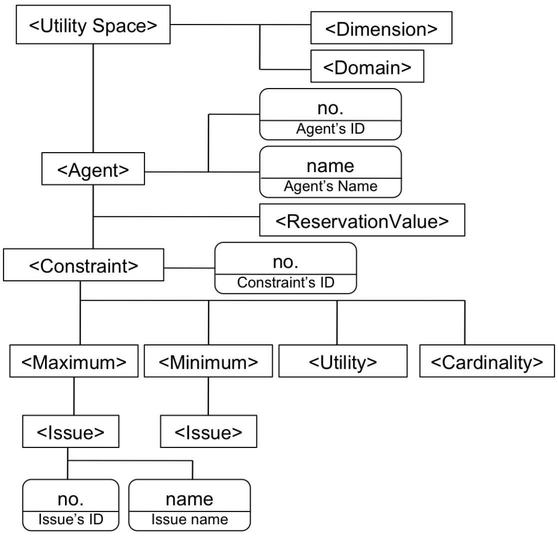

Figure 9.4: Tree-structured XML chart for cube-constraints

XML format for cube-constraints: Figure 9.3 shows an example of the XML format for cube-constraints. Figure 9.4 shows a tree-structured chart for cube-constraints. The tree-structured chart enables us to understand the parent-child relation between elements. A detailed description of the elements is described as follows:

<UtilitySpace>: Utility Space element shows the specification information about the entire utility space. This element has the subelements of "Dimension", "ValueNumber", and "Agent".

<Dimension>: This element specifies the number of issues. In Figure 9.3, the number of issues is four.

<Domain>: This element specifies the value domain for each issue. In Figure 9.3, the domain of all issues is 0,. . . 9.

<Agent>: This element, which specifies the agents, has attributes of agent's id and name. In Figure 9.3, the agent's id is 0 and its name is Alice. There could be multiple agent elements in UtilitySpace element. This element has the subelements of ReservationValue and many Constraint elements.

<ReservationValue>: This element specifies the reservation utility value for determining whether to "agree" or "disagree" with the contract alternatives in a negotiation. In Figure 9.3, the reservation value is 21.

<Constraint>: This element, which defines the constraints, has the id of the constraint as an attribute. This element has the subelements of Issue, Utility, and Cardinality. In Figure 9.3, the id of the constraints is 0.

<Minimum>: This element defines the possible minimum values for each issue. In Figure 9.3, the possible minimum value of Issue 2 is 4. This means that the value for the issue should have more than 4.

<Maximum>: This element defines the possible maximum values for each issue. In Figure 9.3, the possible maximum value of Issue 2 is 8. This means that the value for the issue should have less than 8.

<Utility>: This element defines the utility value in this constraint. The constraints have this utility value if the value for each issue is in the range defined by Issue elements. In Figure 9.3, constraint 0 has a value of 69, and it holds if the value for Issue 1 is 0, the value for issue 2 is 8, the value for Issue 3 is in the range $[4, 8]$, and the value for Issue 4 is 4.

<Cardinality>: This element shows the number of issues related to this constraint. In Figure 9.3, the cardinality is one. This is because this constraint is related to issue 2. In the other words, this constraint is constrained by a issues. In our definition, the contract has a value if only the issues related to the constraints satisfy the possible values. In other words, all values are permitted in other issues not related to the constraint.

XML formats for cone-constraints: Figure 9.5 shows an example of an XML for cone-constraints. Figure 9.6 shows a tree-structured chart for cone-constraints. The XML elements in the "UtilitySpace" and "Agents" elements are almost the same as the XML elements for cube-constraints. A detailed description of the elements in the cone-based constraints is described as follows:

```xml
<?xml version="1.0" encoding="Shift_JIS" standalone="no"?>
<UtilitySpace>
<Dimension>5</Dimension>
<Domain>0-10</Domain>
<Agent name="Alice" no="0">
<ReservationValue>21</ReservationValue>
<Constraint no="0">
<Cardinality>1</Cardinality>
<MaxUtility>122</MaxUtility>
<RiskAttitude>1</RiskAttitude>
<CenterPoint>
<Issue name="4" no="4">0</Issue>
</CenterPoint>
<Width>
<Issue name="4" no="4">2</Issue>
</Width>
</Constraint>
<Constraint no="1">
<Cardinality>1</Cardinality>
...
</Constraint>
</Agent>
</UtilitySpace>
```


Figure 9.5: Cone-constraints XML

<MaxUtility>: This element shows the central value, which is the highest utility in the constraint. In Figure 9.5, the central value is 122, which is the maximum utility in the constraint.

<RiskAttitude>: This element shows a gradient function that represents the risk attitude for making agreements. In our testbed generating tool, we defined a gradient function for each number. For example, one is defined that a gradient function constant is constant. In Figure 9.5, the risk attitude for making agreements is one. Future work includes an extension that enables users to simply define the gradient function.

<Width>: This element shows the impact region, which represents the region affected by the constraint. The impact region is defined in each Issue element. In Figure 9.5, the impact region in Issue 4 is two.

<CenterPoint>: This element shows the central point, where the utility is maximum. In the CenterPoint element, the central point is defined by Issue elements. In Figure 9.5, the central point is 0 in Issue 4 and all values are permitted in other issues (Issues 0 - 3).

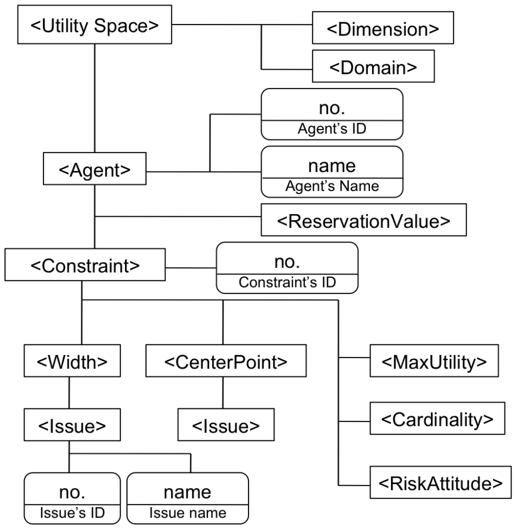

Figure 9.6: Tree-structured chart for cone-constraints XML

9.3 Java program using the testbed

In this subsection, we describe the Java program using the testbeds proposed in the previous section. Our code was implemented in Java 2 (1.5).

Figure 9.7 shows the flow of the JAVA program using testbeds. This program inputs XML files generated by the tool. The following are the details of this program behavior:

Analyzing XML files: In this program, an XML file is analyzed by a Document Object Model (DOM) [144], which is a platform and a language-independent standard object model for representing HTML or XML documents as well as an Application Programming Interface (API) for querying, traversing, and manipulating such documents. The information of the structure of the utility space and the agent's utility function are read from XML files.

Defining the utility function for each agent The structure of the utility space and the agent's utility function are defined based on the XML analyzed in the previous step.

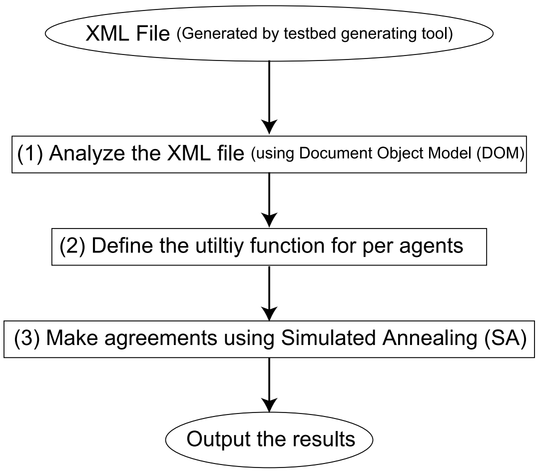

Figure 9.7: Program flow using testbeds

Searching agreements using SA In this program, we provide a simple agreement algorithm that gathers and aggregates all individual agent's utility spaces into one central place and then finds the most optimal contract using simulated annealing (SA) [123]. In simulated annealing, the mediator moves randomly if the temperature is high, but he/she moves to the highest neighbor if the temperature is low. A simulated-annealing method of making agreements was employed in previous works [56] because this search method is superior to other search methods, such as hill climbing search in multi interdependent issue negotiation.

In future work, we will generate this program using other programming languages such as C++, Ruby, Python, and Perl so that this testbed can be used by many users.

10

Conclusion and Future Works

10.1 Conclusion

The work described in this thesis makes a number of important contributions to the state of the art in the area of automated negotiation. The contributions of this work can be summarized as follows:

Chapter 3: We described 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. Constraint-based utility function produces a “bumpy” and highly nonlinear utility function. In 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. These techniques make bid generation computationally tractable even in large utility spaces. A mediator then finds a combination of bids that maximizes social welfare.

Chapter 4: We proposed a threshold adjusting mechanism in very complex negotiations among software agents. We assumed the negotiation with interdependent issues, in which agent utility functions are nonlinear. Many real-world negotiation problems are complex ones involving interdependent multiple issues. We proposed the reveled area which represent the amounts of agent's revealed utility information. Moreover, threshold adjusting mechanism reduces agent's revealed private information. Additionally, this mechanism can reduce the computational cost for finding the deal with high optimality. The experimental results demonstrated that the threshold adjusting mechanism can reduce the computational cost and has enough optimality.

Chapter 5: 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.

Chapter 6: We proposed a nonlinear utility function based on cone-constraints and proposed the Distributed Mediator Protocol (DMP) that can reach agreements and completely conceal agent's utility information and achieve high scalability in utility space. Moreover, we proposed the Hybrid Secure Protocol (HSP) that combines DMP and Take it or Leave it (TOL) protocol. Experimental results demonstrated that HSP can reduce memory with high optimality in cone-constraints and cube-constraints situations.

Chapter 7: We showed that the Nash Bargaining Solution, although provably optimal for negotiations with linear utilities, can lead to sub-optimal outcomes when applied to nonlinear negotiations. We also presented the Secure and Fair Mediator Protocol (SFMP), a novel negotiation protocol that utilizes a combination of nonlinear optimization, secure information sharing, and an approximated fairness metric, and demonstrate that it achieves higher social welfare values than a protocol based on searching for the Nash bargaining solution. Finally, we demonstrated that SFMP out-performs our own previous efforts to enable multi-lateral negotiations in complex domains.

Chapter 8: We proposed a new negotiation protocol, based on grouping issues, which can find high-quality agreements in interdependent issue negotiation. In this protocol, agents generate their private issue interdependency graphs, the mediator identifies the issue-groups based on these graphs, and multiple independent negotiations proceed for each issue sub-group. We demonstrated that our proposed protocol has greater scalability than previous work, and analyzed the effectiveness of different measures of the interdependency rate.

Chapter 9: We proposed a testbed generating tool based on XML for multi-issue negotiation. Our tool provides a common testbed to evaluate the effectiveness of multi-issue negotiation protocols. Moreover, users can easily understand the meaning of data because it is based on a simple XML format. In this testbed, four types of utility spaces were provided that corresponded to real negotiation cases. Finally, we demonstrated examples of experiments using our testbed in which we analyzed the differences among types of utility spaces.

10.2 Future Works

Whole Future work includes building protocols that can find Pareto-optimal contracts more quickly, making them more scalable, and increasing the fairness performance. One potential approach to this problem is to focus the search efforts of the mediators more closely on the fair portion of the Pareto frontier.

Whole: We plan to investigate incentive-compatibility issues in more detail, to ensure that the protocol can not be "gamed" by agents seeking to gain disproportionate influence or to sabotage the outcomes. What we need is an enhancement of our negotiation protocol that incentivizes truthful bidding, preserving equity and maximizing social welfare. In the bilateral case, we found this can be done using a kind of Clarke tax [128], wherein each agent has a limited budget from which it has to pay other agents before the mediator will accept a contract that favors that agent but reduces utility for the others. This approach gives agents the incentive to avoid exaggeration, because exaggerating will cause them to spend their limited budget on contracts that do not strongly impact their true utility values. We will investigate whether and how this approach can be applied to the multilateral case.

Whole: We will analyze the effectiveness of our automated negotiation protocol in the ordinal and cardinal utilities. In this thesis, we introduced the cardinal utilities to constraint based utility functions, however, there are other utility functions based on the cardinal utilities [113].

Chapter 4 & Chapter 5: In a real parliamentary system, the representatives (in theory) have done their best to model the utility functions of the people they represent, so the solutions that satisfy the representatives are likely to be good for (the majority of) the people they represent. The utility functions of the representatives are purely idiosyncratic to them, so the solutions preferred by the representatives may be different from the solutions that are best for the other agents. Therefore, our approach has difficulty finding the best solution in one-shot negotiation. Changing representatives in multi-round negotiation helps support this because the possibility of selecting the best representatives in multi-round negotiation is higher than in one-shot negotiation. However, the changing mechanism proposed here is simple. Thus investigating changing mechanisms is possible future work. The effect of changing mechanisms on selecting representatives is an especially important analytic point.

Chapter6 & Chapter 7: We plan to explore the consequences of the fact that nonlinear problems, unlike linear ones, can produce situations where you have to decide if social welfare or fairness is more important. We will explore protocols that can deal with this situation somehow, for example for giving negotiators the Pareto front and letting them bargain using traditional iterative concession techniques.

Chapter 8: For future work, we will investigate how to improve optimality while maintaining the failure rate advantages of our protocol. One possible track, for example, is to select the number of issue groups adaptively based on the issue dependency topology. Another is to conduct additional negotiation, after the concurrent sub-contract negotiations, to try to increase the satisfaction of constraints that crossed issue group boundaries and were thus ignored in our issue grouping approach.

References

[1] S. Aknine, S. Pinson, and M. F. Shakun. An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems*, 8(1):5–45, 2004. 30

[2] M. S. Alvard. 14. the ultimatum game, fairness, and cooperation among big game hunters. *Foundations of Human Sociality*, 1(9):413–451, 2004. 92

[3] B. An, V. R. Lesser, D. Irwin, and M. Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proc. of the Ninth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 981–988, 2010. 36

[4] K. J. Arrow. *Social Choice and Individual Values*. Yale Univ. Press, 1951. 7, 26, 60

[5] L. M. Ausubel and P. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1), 2002. 120

[6] M. Barbuceanu and W.-K. Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In *Proc. of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC-2000)*, pages 15–30, 2000. 33

[7] G. E. Bolton. A comparative model of bargaining: Theory and evidence. *American Economic Review*, 81(5):1096–136, 1991. 92

[8] T. Bosse, C. M. Jonker, L. van der Meij, and J. Treur. Automated formal analysis of human multi-issue negotiation processes. *Multiagent and Grid Systems*, 4(2):213–233, 2008. 34

[9] C. Boutilier, R. Das, J. O. Kephart, G. Tesauro, and W. E. Walsh. Cooperative negotiation in autonomic systems using incremental utility elicitation. In *Proceed-*

ings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI-2003), pages 89–97, 2003. 33

[10] C. Boutilier and H. H. Hoos. Bidding languages for combinatorial auctions. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, pages 1211–1217, 2001. 5

[11] C. Cappiello, M. Comuzzi, and P. Plebani. On automated generation of web service level agreements. In J. Krogstie, A. Opdahl, and G. Sindre, editors, *Advanced Information Systems Engineering*, volume 4495 of *Lecture Notes in Computer Science*, pages 264–278. Springer Berlin / Heidelberg, 2007. 2

[12] C. Castelfranchi. Modelling social action for ai agents. *Artif. Intell.*, 103:157–182, August 1998. 16

[13] K. Chatterjee and W. Samuelson. Bargaining under incomplete information. *OPERATIONS RESEARCH*, 31(5):835–851, 1983. 25

[14] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Reaching envy-free states in distributed negotiation settings. In *Proc. of 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1239–1244, 2007. 96

[15] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in -additive domains: preference representation and complexity. *Annals OR*, 163(1):49–62, 2008. 5

[16] S. E. Conry, K. Kuwabara, V. Lesser, and R. Meyer. Multistage negotiation for distributed constraint satisfaction. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(6):1462 –1477, 1991. 21

[17] S. E. Conry, R. A. Meyer, and V. R. Lesser. Multistage negotiation in distributed planning. In A. H. Bond and L. Gasser, editors, *Distributed Artificial Intelligence*, pages 367–384. Morgan Kaufmann Publishers Inc., 1988. 1

[18] P. Cramton, Y. Shoham, and E. Steinberg, Richard. *Combinatorial Auctions.* MIT Press, 2006. 31

[19] J. Dang and M. N. Huhns. An extended protocol for multiple-issue concurrent negotiation. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, pages 65–70. AAAI Press, 2005. 30

[20] J. Debenham. Bargaining with information. In *Proc. of th 3rd Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2004)*, pages 663–670, 2004. 34

[21] V. Denicolo and M. Mariotti. Nash bargaining theory, nonconvex problems and social welfare orderings. *Theory and Decision*, 48(4):351–358, 2000. 9, 91

[22] A. Diniz Da Costa, C. J. Lucena, V. Torres Da Silva, S. C. Azevedo, and F. A. Soares. Art competition: Agent designs to handle negotiation challenges. In *Trust in Agent Societies: 11th International Workshop, TRUST 2008, Estoril, Portugal, May 12 -13, 2008. Revised Selected and Invited Papers*, pages 244–272. Springer-Verlag, 2008. 37

[23] N. Egels-Zandn. Tnc motives for signing international framework agreements: A continuous bargaining model of stakeholder pressure. *Journal of Business Ethics*, 84(4):529–547, 2009. 1

[24] H. Elizabeth, M. Kevin, S. Keith, and S. Vernon. Preferences, property rights, and anonymity in bargaining games. *Games and Economic Behavior*, 7(3):346–380, 1994. 1

[25] S. D. Eppinger. Model-based approaches to managing concurrent engineering. *Journal of Engineering Design*, 2(4):283–190, 1991. 8

[26] S. D. Eppinger, D. E. Whitney, and D. A. Gebala. Organizing tasks in complex design projects: Development of tools to represent design procedures. In *NSF Design and Manufacturing Systems Conference*, pages 301–309, 1992. 8

[27] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation Decision Functions for Autonomous Agents. *International Journal of Robotics and Autonomous Systems*, 24:3–4, 1998. 4, 27

[28] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. In *4th International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 119–126, 2000. 27

[29] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. In *Artificial Intelligence*, pages 142:205–237, 2002. 4, 5, 27

[30] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation of multiple issues in incomplete information settings. In *Proc. of Third International Joint Conference on Autonomous Agent and Multi-Agent Systems (AAMAS-2004)*, pages 1080–1087, 2004. 5

[31] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *Proc. of the 8th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1041–1048, 2007. 35

[32] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Approximate and online multi-issue negotiation. In *Proc. of th 6th Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2007)*, pages 947–954, 2007. 35

[33] R. Fisher. *Getting to Yes : How to Negotiate Agreement Without Giving in (AUDIO CASSETTE).* Sound Ideas, 1987. 1

[34] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, 1983. 19

[35] K. Fujita, T. Ito, H. Hattori, and M. Klein. An approach to implementing a threshold adjusting mechanism in very complex negotiations: A preliminary result. In *Proc. of The 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS-2007)*, 2007. 7, 65

[36] K. Fujita, T. Ito, and M. Klein. Preliminary result on secure protocols for multiple issue negotiation problems. In *Proc. of The 11th Pacific Rim International Conference on Multi-Agents (PRIMA-2008)*, 2008. 95

[37] K. Fujita, T. Ito, and M. Klein. Approximately fair and secure protocols for multiple interdependent issues negotiation. In *Proc. of The Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, pages 1287–1288, 2009. 95

[38] N. Fukuta and T. Ito. Fine-grained efficient resource allocation using approximated combinatorial auctionsa parallel greedy winner approximation for large-scale problems. *Web Intelligence and Agent Systems*, 7(1):43–63, 2009. 31

[39] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation

and trust (art) testbed: experimentation and competition for trust in agent societies. In *The 4th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2005)*, pages 512–518, 2005. 37

[40] N. Gatti, F. D. Giunta, and S. Marino. Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *Artificial Intelligence*, 172(8-9):1119 – 1157, 2008. 35

[41] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein. Cooperation without communication. In *the 5th National Conference on Artificial Intelligence (AAAI-1986)*, pages 51–57, 1986. 26

[42] E. Gerding, D. Somefun, and H. L. Poutre. Efficient methods for automated multi-issue negotiation: Negotiating over a two-part tariff. *International Journal of Intelligent Systems*, 21:99–119, 2006. 35

[43] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(2):167 – 189, 1997. 5

[44] R. Greenstadt, J. Pearce, and M. Tambe. Analysis of privacy loss in distributed constraint optimization. In *Proc. of The 21th Association for the Advancement of Artificial Intelligence (AAAI-2006)*, 2006. 33

[45] A. Greenwald and P. Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing*, 5(2):52–60, 2001. 37

[46] J. Grossklags and C. Schmidt. Software agents and market (in) efficiency: a human trader experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(1):56–67, 2006. 2

[47] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973. 114, 120

[48] S. Gupta and Z. A. Livne. Resolving a conflict situation with a reference outcome: An axiomatic model. *MANAGEMENT SCIENCE*, 34(11):1303–1314, 1988. 25

[49] H. Hattori, M. Klein, and T. Ito. A multi-phase protocol for negotiation with interdependent issues. In *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-2007)*, pages 153–159, 2007. 6, 10

[50] K. Hausken. Game-theoretic and behavioral negotiation theory. *Group Decision and Negotiation*, 6:511–528, 1997. 21

[51] K. V. Hindriks and C. M. Jonker. Creating human- machine synergy in negotiation support systems: Towards the pocket negotiator. In *Proc. of First International Working Conference on Human Factors and Computational Models in Negotiation(HuCom-2008)*, pages 47–55, 2008. 2

[52] K. V. Hindriks, C. M. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1397–1398, 2009. 37

[53] K. V. Hindriks, C. M. Jonker, and D. Tykhonov. Interdependencies between issues for multi-issue negotiation. *Cooperative Information Agents X, Lecture Notes in Computer Science*, 4149:301–316, 2006. 35

[54] K. V. Hindriks, C. M. Jonker, and D. Tykhonov. Eliminating issue dependencies in complex negotiation domains. *Multiagent and Grid Systems*, 6(5-6):477–501, 2010. 35

[55] P. T. Hopmann. *The Negotiation Process and the Resolution of International Conflicts (Studies in International Relations)*. University of south Carolina Press, 1996. 1

[56] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces. In *Proc. of 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1347–1352, 2007. 5, 6, 8, 10, 39, 44, 47, 49, 52, 59, 60, 61, 76, 98, 100, 114, 116, 121, 123, 130

[57] T. Ito, H. Hattori, M. Zhang, and T. Matsuo. *Rational, Robust, Secure Negotiations in Multiagent Systems*. IEEE Computer Society, 2005. 2

[58] T. Ito, M. Klein, and H. Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems*, 4(1):67–83, 2008. 6

[59] T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki. *Rational, Robust, and Secure Negotiations in Multiagent Systems*. Series of Studies in Computational Intelligence, Springer-Verlag, 2008. 2

[60] T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki. *Advances in Agent-Based Complex Automated Negotiations*. Series of Studies in Computational Intelligence, Springer-Verlag, 2010. 2

[61] T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki. *Innovations in Agent-Based Complex Automated Negotiations*. Series of Studies in Computational Intelligence, Springer-Verlag, 2011. 2, 27, 37

[62] N. Jennings, P. Faratin, A. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10:199–215, 2001. 4, 18

[63] Y. Jin and M. Geslin. Roles of negotiation protocol and strategy in collaborative design. In J. S. Gero and A. K. Goel, editors, *Design Computing and Cognition '08*, pages 491–510. Springer Netherlands, 2008. 2

[64] Y. Jin and S.-Y. Lu. Agent based negotiation for collaborative design decision making. *CIRP Annals - Manufacturing Technology*, 53(1):121 – 124, 2004. 2

[65] C. M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15:221–252, 2007. 34, 35

[66] C. M. Jonker and J. Treur. An agent architecture for multi-attribute negotiation. In *Proc. of 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pages 1195–1201, 2001. 35

[67] E. Kalai and M. Smorodinsky. Other solutions to nash's bargaining problem. *Econometrica*, 43(3):513–518, 1975. 24

[68] M. Kaneko. An extension of the nash bargaining problem and the nash social welfare function. *Theory and Decision*, 12(2):135–148, 1980. 9, 91

[69] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives*. Cambridge University Press, 1993. 2, 20, 93

[70] B. Ken, A. Shaked, and J. Sutton. Testing noncooperative bargaining theory: A preliminary study. *American Economic Review*, 75:1178–1180, 1985. 92

[71] A. W. Ken Binmore, Ariel Rubinstein. The nash bargaining solution in economic modeling. *The RAND Corporation in its journal RAND Journal of Economics*, 17:176–188, 1986. 9, 91

[72] G. Kersten and H. Lai. Negotiation support and e-negotiation systems: An overview. *Group Decision and Negotiation*, 16:553–586, 2007. 2

[73] B. Khanal, H. Sugiura, T. Ito, M. Iwasaki, K. Fujita, and M. Kobayashi. *Smart-Contractor*: A distributed task assignment system based on the simple contract net protocol. In *International Conference on Principles of Practice in Multi-Agent Systems (PRIMA-2009)*, pages 403–415, 2009. 30

[74] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):58–73, 2003. 4, 5, 20, 27

[75] M. Klein and L. Iandoli. Supporting collaborative deliberation using a large-scale argumentation system: The mit collaboratorium. In *Proc. of Directions and Implications of Advanced Computing Symposium Conference on Online Deliberation (DIAC-2008/OD2008)*, 2008. 2

[76] G. Koumoutsos and K. Thramboulidis. A knowledge-based framework for complex, proactive and service-oriented e-negotiation systems. *Electronic Commerce Research*, 9:317–349, 2009. 2

[77] S. Kraus. *Strategic Negotiation in Multiagent Environments.* Cambridge University Press, 2001. 2

[78] G. Lai, K. Sycara, and C. Li. A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multiagent Grid Systems*, 4:45–65, January 2008. 34

[79] K. R. Lai and M.-W. Lin. Modeling agent negotiation via fuzzy constraints in e-business. *Computational Intelligence*, 20(4):624–642, 2004. 5

[80] R. K. Lau. Towards genetically optimised multi-agent multi-issue negotiations. In *Proc. of 38th Hawaii International Conference on System Sciences (HICSS-2005)*, 2005. 5

[81] M. Li, Q. B. Vo, and R. Kowalczyk. Searching for fair joint gains in agent-based negotiation. In *Proc. of The 8th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1049–1056, 2009. 36

[82] R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Commun. ACM*, 53(1):78–88, 2010. 2, 36

[83] R. Lin, S. Kraus, Y. Oshrat, and Y. K. Gal. Facilitating the evaluation of automated negotiators using peer designed agents. In *Proc. of The 24th Association for the Advancement of Artificial Intelligence (AAAI-2010)*, 2010. 36

[84] R. Lin, S. Kraus, D. Tykhonov, K. Hindriks, and C. M. Jonker. Supporting the design of general automated negotiators. In *Proceedings of the Second International Workshop on Agent-based Complex Automated Negotiations (ACAN'09)*, 2009. 37

[85] R. Lin, Y. Oshrat, and S. Kraus. Investigating the benefits of automated negotiations in enhancing people's negotiation skills. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 345–352, 2009. 2

[86] R. J. Lin and S. T. Chou. Bilateral multi-issue negotiations in a dynamic environment. In *Proc. of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC-2003)*, 2003. 33

[87] Y. Lindell. *Composition of Secure Multi-Party Protocols: A Comprehensive Study.* Springer, 2003. 79, 81

[88] A. Lomuscio, M. Wooldridge, and N. R. Jennings. A classification scheme for negotiation in electronic commerce. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective.*, pages 19–33. Springer-Verlag, 2001. 2

[89] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings. A classification scheme for negotiation in electronic commerce. *Group Decision and Negotiation*, 12:31–56, 2003. 17

[90] M. Lopez-Carmona, I. Marsa-Maestre, M. Klein, and T. Ito. Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces. *Autonomous Agents and Multi-Agent Systems*, pages 1–51, 2010. 121

[91] M. A. López-Carmona and J. R. Velasco. An expressive approach to fuzzy constraint based agent purchase negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS-2006)*, pages 429–431, 2006. 6

[92] X. Luo, N. R. Jennings, and N. Shadbolt. Acquiring user strategies and preferences for negotiating agents: A default then adjust method. *International Journal of Human Computer Studies*, 64(4):304–321, 2006. 3

[93] X. Luo, N. R. Jennings, N. Shadbolt, H. Leung, and J. H. Lee. A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence*, 148:53–102, 2003. 5

[94] X. Luo, C. Miao, N. Jennings, M. He, Z. Shen, and M. Zhang. Kemnad: A knowledge engineering methodology for negotiating agent development. *Computational Intelligence*, 2011. 36

[95] R. T. Maheswaran, J. P. Pearce, E. Bowring, P. Varakantham, and M. Tambe. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *Autonomous Agents and Multi-Agent Systems*, 13(1):27–60, 2006. 33

[96] T. W. Malone and M. Klein. Harnessing collective intelligence to address global climate change. *Innovations Journal*, 2(3):15–26, 2007. 2

[97] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, T. Ito, K. Fujita, and M. Klein. Balancing utility and deal probability for negotiations in highly nonlinear utility spaces. In *Proc. of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009. 116

[98] I. Marsa-Maestre, M. A. L.-C. nad Juan R. Velasco, and E. de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *Proc. of The 8th Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1057–1064, 2009. 116

[99] T. Matsui, H. Matsuo, and A. Iwata. Efficient method for asynchronous distributed constant optimization algorithm. In *Proceedings of Artificial Intelligence and Applications (AIA2005)*, pages 727–732, 2005. 33

[100] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149 – 180, 2005. 33

[101] R. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, 1991. 94

[102] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265 – 281, 1983. 25

[103] J. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950. 1, 4, 9, 23, 91, 94

[104] J. F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950. 23

[105] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 23

[106] Y.-K. Ng. *Welfare Economics*. Macmillan, 1983. 19

[107] N. Nisan, T. Roughgarden, E. Tardos, and V. V. V. Eds. *Algorithmic Game Theory*. Cambridge University Press, 2007. 23, 31

[108] J. Niu, K. Cai, S. Parsons, P. McBurney, and E. Gerding. What the 2007 tac market design game tells us about effective auction mechanisms. *Journal of Autonomous Agents and Multi-Agent Systems*, 2010. 37

[109] D. C. Parkes. *Online mechanisms. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors, Algorithmic Game Theory, chapter 16.* Cambridge University Press, 2007. 31

[110] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *JOURNAL OF LOGIC AND COMPUTATION*, 8:261–292, 1998. 28

[111] H. V. Parunak. Manufacturing experience with the contract net. *AI MAGAZINE*, 8(2), 1987. 30

[112] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of the 19th international joint conference on Artificial intelligence(IJCAI2005)*, pages 266–271, 2005. 33

[113] H. Raiffa. *The art and science of negotiation.* Harvard University Press, 1982. 1, 4, 134

[114] F. Ren, M. Zhang, C. Miao, and Z. Shen. A market-based multi-issue negotiation model considering multiple preferences in dynamic e-marketplaces. In *Proc. of 12th International ConferencePrinciples of Practice in Multi-Agent Systems (PRIMA-2009)*, pages 1–16, 2009. 2

[115] J. Robertson and W. Webb. *Cake-Cutting Algorithms: Be Fair If You Can.* A K Peters Ltd, 1998. 10, 92, 95, 96

[116] V. Robu. *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*, volume 2009-1. Eindhoven University of Technology, July 2009. 20

[117] V. Robu and H. L. Poutre. Retrieving the structure of utility graphs used in multi-item negotiation through collaborative filtering of aggregate buyer preferences. In *Proc. of The 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)*, 2006. 34

[118] V. Robu, D. J. A. Somefun, and J. L. Poutre. Modeling complex multi-issue negotiations using utility graphs. In *Proc. of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, 2005. 34

[119] J. S. Rosenschein and M. R. Genesereth. Deals among rational agents. In A. H. Bond and L. Gasser, editors, *Distributed Artificial Intelligence*, pages 227–234. Morgan Kaufmann Publishers Inc., 1988. 26

[120] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinational auctions. *Management Science*, 44(8):1131–1147, 1998. 31

[121] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982. 1, 4, 21, 24, 25, 37

[122] A. Rubinstein. A bargaining model with incomplete information about time preferences. *Econometrica*, 53(5):1151–1172, 1985. 25

[123] S. J. Russell and P. Norvig. *Artificial Intelligence : A Modern Approach.* Prentice Hall, 2002. 79, 81, 84, 97, 98, 99, 112, 114, 117, 130

[124] T. Sandholm. Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *Autonomous Agents and Multi-Agent Systems*, 3:73–96, 2000. 2

[125] T. Sandholm and C. Boutilier. *Preference Elicitation in Combinatorial Auctions.* MIT Press, 2006. 3

[126] T. Sandholm and V. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. *First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 328–335, 1995. 2, 30

[127] T. Sandholm and N. Vulkan. Bargaining with deadlines. In *Proc. of The 13th Association for the Advancement of Artificial Intelligence (AAAI-1999)*, pages 44–51, 1999. 25

[128] T. W. Sandholm. *Distributed Rational Decision Making.* Multi-Agent Systems. G. Weiss, Editor, 1998. 133

[129] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of multiobjective optimization.* Academic Press, 1985. 32

[130] S. Sen and E. H. Durfee. The role of commitment in cooperative negotiation. *International Journal on Intelligent Cooperative Information Systems*, 3:67–81, 1994. 30

[131] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. 9, 79, 91, 95

[132] C. Shi, J. Luo, and F. Lin. A multi-agent negotiation model applied in multi-objective optimization. In *Proc. of 9th International ConferencePrinciples of Practice in Multi-Agent Systems (PRIMA-2006)*, pages 305–314, 2006. 32

[133] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2008. 33

[134] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Computers*, 29(12):1104–1113, 1980. 1, 29

[135] L.-K. Soh and X. Li. Adaptive, confidence-based multiagent negotiation strategy. In *Proc. of Third International Joint Conference on Autonomous Agent and Multi-Agent Systems (AAMAS-2004)*, pages 1048–1055, 2004. 5

[136] P. Stone and A. Greenwald. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research*, 5(2):229–265, 2005. 37

[137] K. P. Sycara. Arguments of persuasion in labour mediation. In *Proceedings of the 9th international joint conference on Artificial intelligence (IJCAI-1985)*, pages 294–296, 1985. 29

[138] K. P. Sycara. Resolving goal conflicts via negotiation. In *the 7th National Conference on Artificial Intelligence (AAAI-1988)*, pages 245–250, 1988. 29

[139] K. P. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the 9th international joint conference on Artificial intelligence (IJCAI-1989)*, pages 517–523, 1989. 29

[140] H. Tamura and Y. Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983. 8, 109

[141] W. Thomson. Cooperative models of bargaining. *Handbook of Game Theory with Economic Applications*, 2:1238–1284, 1992. 96

[142] H. R. Varian. *Intermediate microeconomics : a modern approach.* W.W. Norton, New York, 1987. 20

[143] W. Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961. 31

[144] W3C. Document Object Model (DOM) Technical Reports. `http://www.w3.org/DOM/DOMTR`. 122, 129

[145] L. R. Weingart, M. J. Prietula, E. B. Hyder, and C. R. Genovese. Knowledge and the sequential processes of negotiation: A markov chain analysis of response-in-kind. *Journal of Experimental Social Psychology*, 35(4):366 – 393, 1999. 21

[146] M. P. Wellman, A. Greenwald, and P. Stone. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2007. 2

[147] M. P. Wellman, P. R. Wurman, K. O'Malley, R. Bangera, S. de Lin, D. Reeves, and W. E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001. 37

[148] G. Werner, S. Rolf, and S. Bernd. An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior and Organization*, 3:367–388, 1982. 73, 93

[149] A. P. Wierzbicki, L. Krús, and M. Makowski. The role of multi-objective optimization in negotiation and mediation support. *Theory and Decision*, 34:201–214, 1993. 32

[150] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995. 15

[151] R. Yager. Multi-agent negotiation using linguistically expressed mediation rules. *Group Decision and Negotiation*, 16:1–23, 2007. 35

[152] M. Yokoo. *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*. Springer-Verlag, 2001. 32

[153] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3:185–207, June 2000. 32

[154] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name declarations in mechanism design: Towards collective decision making on the internet. In *International Conference on Distributed Computing Systems,*, page 146, 2000. 31

[155] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in internet auctions. *Games and Economic Behavior*, 46:174–188, 2004. 31

[156] D. Zhang. Axiomatic characterization of task oriented negotiation. In *Proc. of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 367–372, 2009. 36

[157] X. Zhang, V. Lesser, and R. Podorozhny. Multi-dimensional, multistep negotiation. *Autonomous Agents and Multi-Agent Systems*, 10:5–40, 2005. 4

[158] E. Zurel and N. Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC2001)*, pages 125–136, 2001. 31

Publications

Awards

1. The 2010 IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT2010) Nominated for Best Paper Award, Sep., 2010

2. Yamashita Research Award, Information Processing Society of Japan (IPSJ), 2010,

   - "Interdependency Rate and Scalable Protocol for Multiple Interdependent Issues Negotiation", SIG Technical Reports (ICS-159), Aug., 2010

3. Winner (The 1st Place) of the Automated Negotiating Agent Competition (ANAC2010) at AAMAS2010, May, 2010.

4. Certified Master Thesis by IPSJ, May 14, 2010

5. Student Encouragement Award of the 72th IPSJ National Convention, March. 15, 2008

6. The President of Nagoya Institute of Technology Award 2009, Feb. 16, 2010

7. Students Research Encouraging Award of The Tokai Regional Section of The Institute of Electronics, Information and Communication Engineers (IEICE) ,2009.

8. Students Research Encouraging Award of The Tokai Regional Section of The Information Processing Society of Japan (IPSJ), Katsuhide Fujita,

"Development of Consensus Mechanism in Multi-issue Negotiation Problems", 2009.

9. The 24th TELECOM System Technology Award for Student: Katsuhide Fujita, "Proposing Consensus Mechanisms in Multi-issue Negotiation Problems."

    - 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),2008.

10. IEEE Student Encouragement Award

    - Katsuhide Fujita, and Takayuki Ito, "A Proposal on A Negotiation Mechanism based on Distributed Mediators for Multi-Issue Negotiation Problems", The Tokai Regional Section of The Institute of Electronics, Information and Communication Engineers , 2009.

11. The President of Nagoya Institute of Technology Award 2007, March, 2008

12. Best Student Paper Award, Nov. 7, 2007.

    - Katsuhide Fujita and Takayuki Ito, "An Approach to Implementing A Threshold Adjusting Mechanism in Very Complex Negotiations: A Preliminary Result", In The Proceedings of The 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS2007), 2007

Journal

1. 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.

2. 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.

3. Katsuhide Fujita, Takayuki Ito and 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.

4. Katsuhide Fujita, Takayuki Ito,and 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.

5. 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.

6. 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.

7. 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.

8. 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

9. 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

International Conference

1. Katsuhide Fujita, Takayuki Ito, Mark Klein, "An Approach to Scalable Multi-issue Negotiation: Decomposing the Contract Space based on Issue Interdependencies", The 2010 IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT2010), Canada, Sep. 2010.(Acceptance rate : 18.8%).

2. Katsuhide Fujita, Takayuki Ito and Mark Klein, "The Effect of Grouping Issues in Multiple Interdependent Issues Negotiation based on Cone-Constrain" The 3rd International Workshop on Agent-based Complex Automated Negotiations (ACAN2010), Tronto, Canada, May 2010.

3. Takayuki Ito, Rafik Hedfi, and Katsuhide Fujita, "Towards Collective Collaborative Design : An Implementation of Agent-mediated Collaborative 3D Products Design System", International Workshop on Multi-Agent Systems and Collaborative Technologies (I-MASC2010), May 2010.

4. Bipin Khanal, Hideyuki Sugiura, Takayuki Ito, Masashi Iwasaki, Katsuhide Fujita, Masao Kobayashi:, "SmartContractor: A Distributed Task Assignment System Based on the Simple Contract Net Protocol." , The 12th International Conference on Principles of Practice In Multi-Agent Systems (PRIMA 2009), pp. 403-415, Nagoya, Japan, Dec. 2010.

5. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Secure and Fair Negotiation Protocol in Highly Complex Utility Space based on Cone-Constraints", The 2009 IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT2009), pp. 427-430, Milan, Italy, Sep. 2009.

6. Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, Takayuki Ito, Mark Klein, Katsuhide Fujita, "Balancing Utility and Deal Probability for Negotiations in Highly Nonlinear Utility Spaces", The Twenty-First

International Joint Conference on Artificial Intelligence (IJCAI2009), pp. 214-219, USA, July 2009.

7. Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Takayuki Ito, Mark Klein, Katsushide Fujita and Juan R. Velasco, “Constraint and Bid Quality Factor for Bidding and Deal Identification in Complex Automated Negotiations” The 2nd International Workshop on Agent-based Complex Automated Negotiations (ACAN2009), Budapest, Hungary, May 2009.

8. Katsuhide Fujita, Takayuki Ito, Mark Klein, “Common Testbed Generating Tool based on XML for Multiple Interdependent Issues Negotiation Problems”, The 2nd International Workshop on Agent-based Complex Automated Negotiations (ACAN2009), Budapest, Hungary, May 2009.

9. Katsuhide Fujita, Takayuki Ito, Mark Klein, “Approximately Fair and Secure Protocols for Multiple Interdependent Issues Negotiation”, The Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2009), pp.1287-1288, Budapest, Hungary, May 2009.

10. Katsuhide Fujita, Takayuki Ito and Mark Klein, “Finding Nash Bargaining Solutions for Multi-issue Negotiations : A Preliminary Result”, The 1st International Working Conference on Human Factors and Computational Models in Negotiation (HuCom2008), Delft, the Netherlands, Dec. 2008.

11. Katsuhide Fujita, Takayuki Ito, and Mark Klein, “Preliminary Result on Secure Protocols for Multiple Issue Negotiation Problems”, The 11th Pacific Rim International Conference on Multi-Agents (PRIMA 2008), pp. 161-172, Hanoi, Viet Nam, Dec. 2008.

12. Katsuhide Fujita, Takayuki Ito, Mark Klein, “Effects of Revealed Area based Selection Method for Representative-based Protocol”, The 1st International Workshop on Agent-based Complex Automated Negotiations (ACAN2008), Estoril, Portugal, May 2008.

13. Katsuhide Fujita, Takayuki Ito, Mark Klein, “A Preliminary result on a representative-based multi-round protocol for multi-issue negotiations”, The

Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2008), pp. 1573-1576, Estoril, Portugal, May 2008.

14. Katsuhide Fujita, Takayuki Ito, "A Preliminary Analysis of Computational Complexity of the Threshold Adjusting Mechanism in Multi-Issue Negotiations", The 3rd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS2007), pp.383-386, California, USA, Nov. 2007.

15. Katsuhide Fujita, Takayuki Ito, Mark Klein, Hiromitsu Hattori "An Approach to Implementing A Threshold Adjusting Mechanism in Very Complex Negotiations: A Preliminary Result", The 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS2007), No.20, Ishikawa Japan, Nov. 2007

Domestic Conference (in Japanese)

1. Katsuhide Fujita, Takayuki Ito, Nobuyasu Mizutani, Mark Klein, "A Consensus Mechanism based on Adjustment of Issue-Groups for Multi-issue Negotiation", 24th JSAI Annual Conference, The Japanese Society for AI(JSAI), 2010.

2. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Consensus Mechanism based on Adjustment of Issue-Groups for Multi-issue Negotiation", Proceedings of the 2010 IEICE General Conference.

3. Katsuhide Fujita, Takayuki Ito, Mark Klein, "Interdependency Rate and Scalable Protocol for Multiple Interdependent Issues Negotiation", SIG Technical Reports (ICS-159), 2010.

4. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A negotiation protocol based on issue-interdependency for highly complex scenario", In the Proc. of the 72th Annual Conference of IPSJ, 2010.

5. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A consensus mechanism based on issue dependency for multiple interdependent issues negotiations", Joint Agent Workshop and Symposium(JAWS2009), 2009.

6. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A consensus mechanism based on issue dependency for complex multiple interdependent issues negotiations", Technical Committee on Artificial Intelligence and Knowledge Processing (AI), 2009.

7. Katsuhide Fujita, Takayuki Ito, Mark Klein, "Common Testbed Tool based on XML for Multiple Issues Negotiation Problems", Forum on Information Technology(FIT2009), 2009.

8. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Utility Model based on Risk Attitude for Multiple interdependent Issues Negotiations", 23th JSAI Annual Conference, The Japanese Society for AI(JSAI), 2009.

9. Katsuhide Fujita, Takayuki Ito, "Finding the nash bargaining solution for multi-issue negotiation", In the Proc. of the 71th Annual Conference of IPSJ, 2009.

10. Katsuhide Fujita, Takayuki Ito, "An Analysis of Pareto Optimality in Distributed Mediator Protocol for Multi Interdependent Issue Negotiation Problems", Joint Agent Workshop and Symposium (JAWS2008), 2008.

11. Katsuhide Fujita, Takayuki Ito, "Hybrid Secure Protocol for Multi-issue Negotiation Problems for Nonlinear Problems", Technical Committee on Artificial Intelligence and Knowledge Processing (AI), 2007.

12. Katsuhide Fujita , Takayuki Ito, "A Proposal on A Negotiation Mechanism based on Distributed Mediators for Multi-Issue Negotiation Problems", The Tokai Regional Section of The Institute of Electronics, Information and Communication Engineers , 2009.

13. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Preliminary Result on Secure and Scalable Protocols for Multiple Issue Negotiation Problems", 22th JSAI Annual Conference, The Japanese Society for AI(JSAI), 2008.

14. Katsuhide Fujita, and Takayuki Ito, "An Analysis on the representative selection method for persuasive negotiation mechanism in multi-issue negotiations" In the Proc. of the 70th Annual Conference of IPSJ, 2008.

15. Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Persuasion Mechanism based on Revealed Private Utility Space In Multi Issue Negotiations", Technical Committee on Artificial Intelligence and Knowledge Processing (AI), 2007.

16. Katsuhide Fujita, Takayuki Ito, Hiromitsu Hattori, "Using Autonomous Threshold Adjustment to Enable Multi- Party Negotiations with Multiple Interdependent Issues", Joint Agent Workshop and Symposium (JAWS2007), 2007.

17. Katsuhide Fujita, Takayuki Ito, "Using Autonomous Threshold Adjustment to Enable Multi-Party Negotiations with Multiple Interdependent Issues", Forum on Information Technology(FIT2007), 2007

Others

1. The 9th IEEE/ACIS International Conference on Computer and Information Science(ICIS 2010) Registration Desk Chair, Special Session Chair, Special Session Chair, Program Committee

2. The Third International Workshop on Agent-based Complex Automated Negotiation (ACAN2010) Program Committee

3. International Workshop on Agent-based Collaboration, Coordination, and Decision Support (ACCDS 2009): Program Chair

4. The 12th International Conference on Principles of Practice In Multi-Agent Systems (PRIMA 2009): Volunteer Leader