Journal of Intelligent and Fuzzy Systems 2009

SECURE AND EFFICIENT PROTOCOLS FOR MULTIPLE INTERDEPENDENT ISSUES NEGOTIATION

Katsuhide Fujita[*]
Department of Techno-Business Administration, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya, Aichi, 466-8555, JAPAN

Takayuki Ito[†] *and Mark Klein*
Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology, Three Cambridge Center, NE20-336, Cambridge, MA 02142, U.S.A.

ABSTRACT

Multi-issue negotiation protocols represent a promising field since most negotiation problems in the real world involve multiple issues. Our work focuses on negotiation with multiple interdependent issues in which agent utility functions are nonlinear. Firstly, we define utility function based on cone-constraints which are nonlinear. The utility function based on cone-constraint is more realistic than existing utility models and configures the risk attitudes to the cone-constraint. However, if the utility function has cone-constraint features, the utility space becomes extremely nonlinear, making it very difficult to find the optimal agreement point. Existing works have not yet concerned with agents' private information that should be concealed from others in negotiations. In this paper, we propose Distributed Mediator Protocol and Take it or Leave it Protocol for negotiation that can reach agreements and completely conceal agents' private information. Moreover, we propose Hybrid Secure Protocol that combines Distributed Mediator Protocol with Take it or Leave it Protocol. The

[*]Tel:+81-52-735-7968, Fax:+81-52-735-7404, E-mail:fujita@itolab.mta.nitech.ac.jp
[†]Visiting from Nagoya Institute of Technology

Hybrid Secure Protocol can also reach agreements while completely concealing agents' private information. Furthermore, the Hybrid Secure Protocol achieves high optimality and uses less communication cost. We demonstrate the performance of Hybrid Secure Protocol in cone-constraints and cube-constraints situations.

(1) INTRODUCTION

Multi-issue negotiation protocols represent an important field of study. Even though there has been a lot of previous work in this area ([2, 3, 4] etc.), most have dealt exclusively with simple negotiations involving independent multiple issues. Many real-world negotiations, however, are complex and involve interdependent multiple issues. First, we propose a model of highly complex utility spaces based on "cone-constraints". We use cone-constraints to capture the intuition that agents' utilities for a contract usually decline gradually, rather than step-wise, with distance from their ideal contract. In previous works, the utility model of cube-constraints is focused in multiple interdependent issues negotiation field([7, 11] etc.). The model of cone-constraints is more realistic than existing works such as cube-constraints on multiple interdependent issues negotiation. These previous studies mainly assume that agents have an incentive to cooperate to achieve win-win agreements because the situation is not a zero-sum game.

Existing works have not yet been concerned with agents' private information. In negotiation, agents' private information should not be revealed to other agents and mediators. For example, suppose that several companies collaboratively design and develop a new car model. If one company reveals more private information than the other companies, the other companies will know more of that company's important information, such as utility information. As a result, the company suffers a disadvantage in subsequent negotiations, and the mediator might leak the agent's utility information. Furthermore, explicitly revealing private information is dangerous for privacy reasons. Therefore, our aim is to create a protocol that will find high-quality solutions while concealing agent's private information.

We previously proposed a bidding-based negotiation protocol that focuses on interdependent multiple issues. Agents generate bids by sampling and searching their utility functions, and the mediator finds the optimum combination of submitted bids from agents [11]. This protocol can achieve an agreement without revealing all agent private information. Moreover, we proposed a threshold adjusting mechanism where the mediator adjusts the agent's threshold for generating bids. In the threshold adjusting mechanism, agents make agreements without excessively revealing their utility information [6]. However, since in these protocols the computational complexity for finding the solution is too large, we proposed a representative-based protocol [8] where the mediator selects representatives who propose alternatives to other agents. This protocol drastically reduced the computational complexity of the number of agents because the number of agents that make agreements was reduced.

Though, there are two main issues in the above protocols. First, it is impossible for the above protocols to conceal all agent private information because agents have to reveal some private information. Additionally, scalability for the complexity of agent's utility function isn't very high. Especially, if the utility function has cone-constraint features, the utility space becomes extremely nonlinear, making it very difficult to find the optimal agreement point. Therefore, we need to create another new protocol that conceals all agent private information with high scalability for the complexity of agent utility functions.

In this paper, 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 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, a simulated annealing and a genetic algorithm in cone-constraint and cube-constraint situations. Hill climbing and simulated annealing have been employed in the previous works[7, 11]. However, genetic algorithm also performs well to find high optimal contract. Therefore, we compare GA-based method with the other methods in this paper. Additionally experiment results in previous works[7] is only evaluated in cube-constraints. In this paper, we evaluate the methods in cone-constraints situation that is highly complex.

The remainder of the paper is organized as follows. First, we describe a model of nonlinear multi-issue negotiation and utility function based on cube-constraints and cone-constraints. Second, we propose the Distributed Mediator Protocol (DMP) and the Take it or Leave it (TOL) Protocol. Third, we propose the Hybrid Secure Protocol (HSP). Fourth, we present the experimental results about optimality and communication cost (memory). Finally, we describe related works and draw conclusions.

(2) NONLINEAR UTILITY FUNCTION

In the literature of multi-issue negotiations, we consider the situation where n agents want to reach an agreement with a mediator who manages the negotiation from the middle position. There are m issues, $s_j \in S$, to be negotiated. The number of issues represents the number of utility space dimensions. For example, if there are three issues, the utility space has three dimensions. The issues are not "distributed" over agents, who are all negotiating a contract with N (e.g. 10) issues in it. All agents are potentially interested in the values for all N issues. Issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in [0, X](1 \leq j \leq M)$. A contract is represented by a vector of issue values $\vec{s} = (s_1, ..., s_m)$. The objective function for agreement search protocols can be described as follows:

$$\arg\max_{\vec{s}} \sum_{i \in N} u_i(\vec{s}).$$

The proposed protocols in the literature try to find contracts that maximize social welfare, *i.e.*, the total utilities for all agents. Such contracts, by definition, will also be Pareto-optimal.

In this paper, we deal with cube-constraints and cone-constraints as the utility function. Every agent has its own, typically unique, set of constraints.

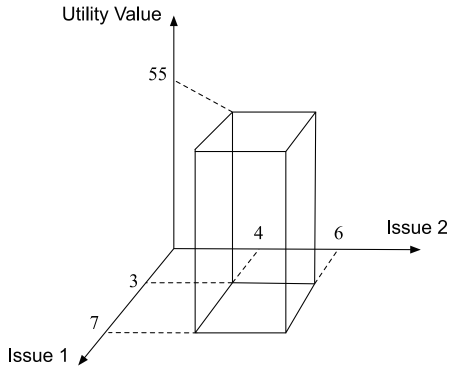

Figure 1: Example of a cube-constraint

Cube-constraints: An agent's utility function is described in terms of constraints [11]. There are l constraints, $c_k \in C$. Each constraint represents a region with one or more dimensions and has an associated utility value. Constraint c_k has value $w_i(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}(1 \leq k \leq l)$. We call this type of constraint a "cube"-constraint. Figure 1 shows an example of a binary constraint between Issues 1 and 2. This constraint, which has a value of 55, holds if the value for Issue 1 is in the range $[3, 7]$ and the value for Issue 2 is in the range $[4, 6]$.

In recent works (e.g. [10]), several types of cube-constraints were proposed.

Cone-constraints: An agent's utility function can be described in terms of cone-constraints. Figure 2 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

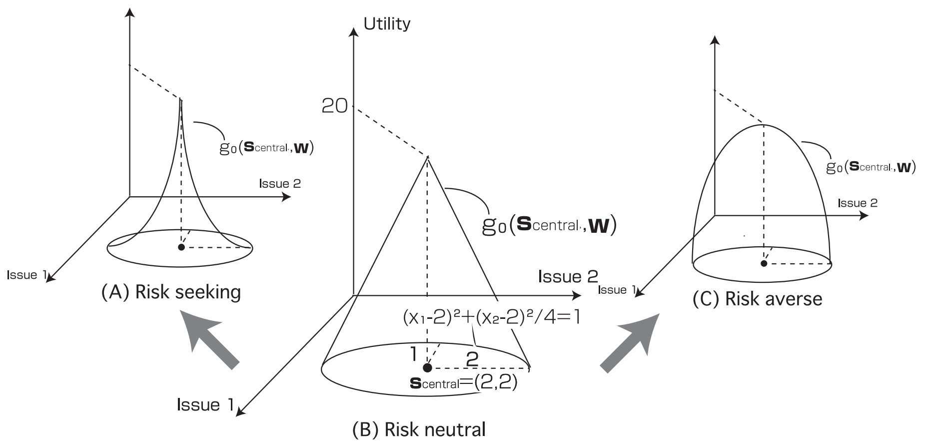

Figure 2: Example of cone-constraints

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 $\vec{w}$, 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 $\vec{s}$. In this paper, 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 2 (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

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

(A) in Figure 2 (e.g., exponent). If an agent has a risk-averse attitude for c_k, g_k is defined as (C) in Figure 2. 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.

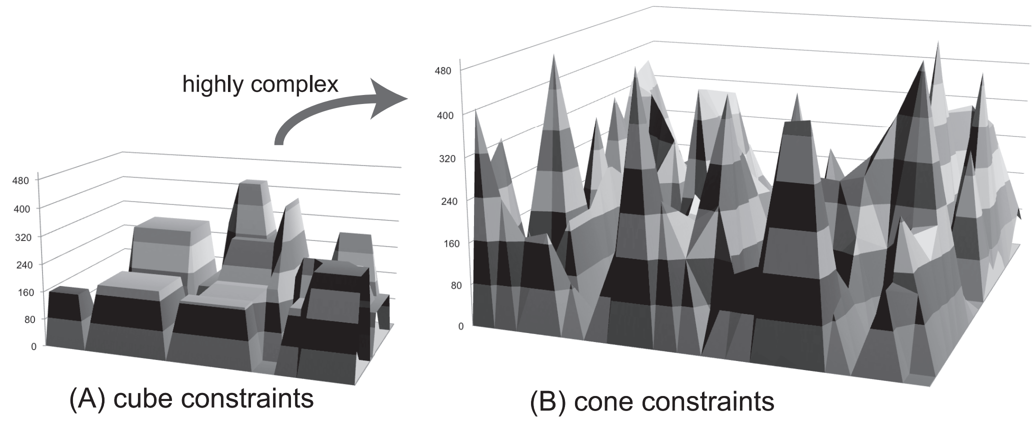

Figure 3: Example of utility space

Figure 3 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.

(3) SECURE NEGOTIATION PROTOCOLS

(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[16, 21]. DMP is shown as follows.

We assume n mediators ($M_0, \ldots, M_n$) who can calculate the sum of all the agent utility values if k mediators get together, and there are m 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 [20]. 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 $v_{1,j}, \ldots, v_{m,j}$ from all agents. M_j calculates $v_j = v_{1,j} + \cdots + v_{m,j}$ *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 4 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 ($k = n$). 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.

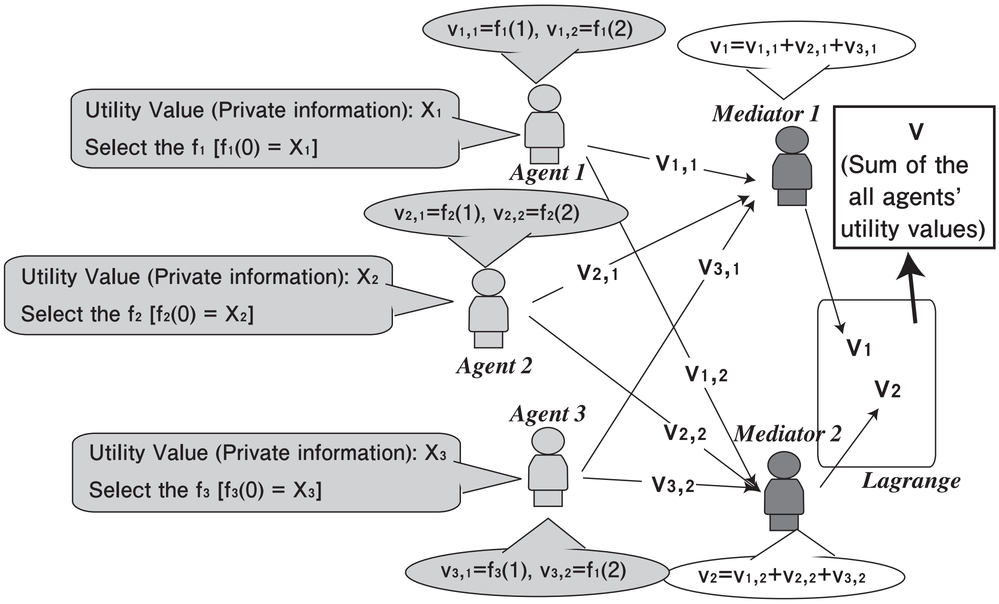

Figure 4: Distributed Mediator Protocol

Privacy DMP can calculate the sum of all agent utility values and conceal them. The proof is identical to the Multi-Party Protocol [16]. 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.

(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 [20], 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.

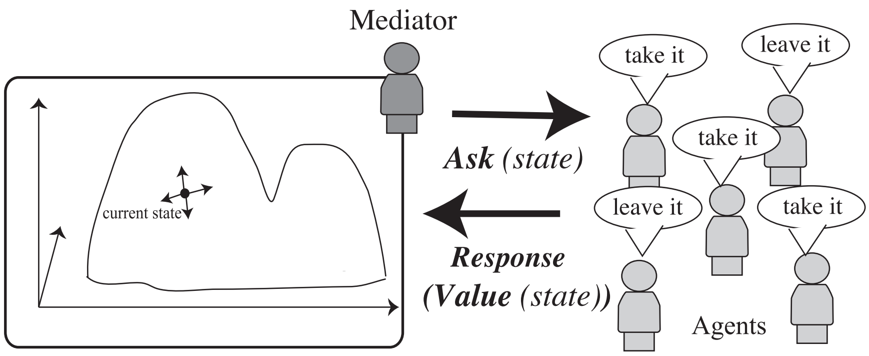

Figure 5: Take it or Leave it (TOL) Protocol

Figure 5 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.

(4) HYBRID SECURE PROTOCOL (HSP) FOR NEGOTIATION

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 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 $\sim$ step 5 in DMP proposed in 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.

(5) EXPERIMENTAL RESULTS

(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 [20]. "(B) DMP (HC)" is the Distributed Mediator Protocol and the search algorithm is hill-climbing [20]. "(C) DMP (GA)" is the Distributed Mediator Protocol and the search algorithm is the genetic algorithm [20]. "(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.

(5.2) Experimental Results

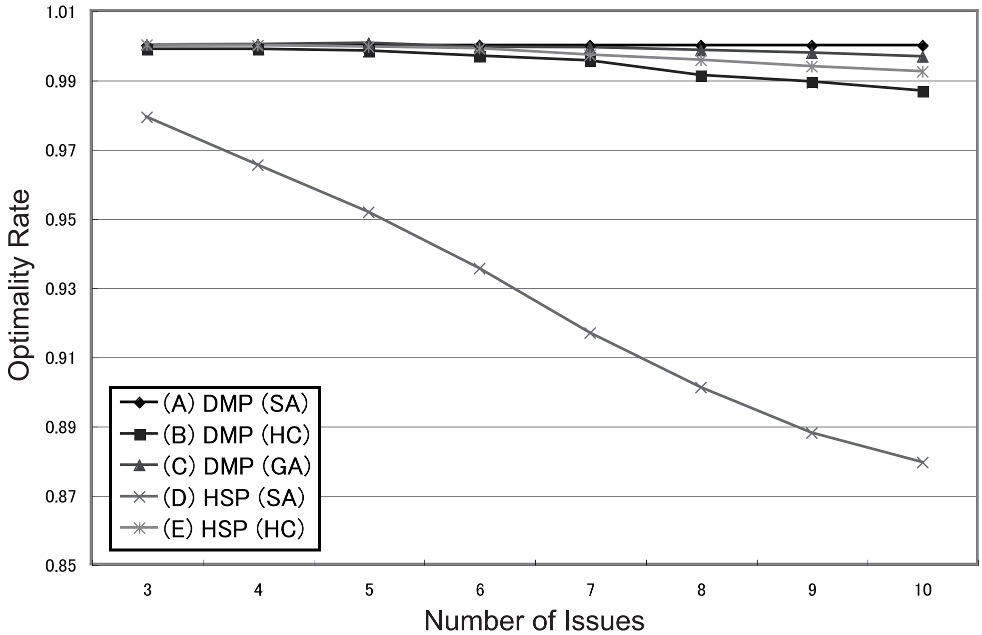

Figure 6: Optimality Rate (Cube-constraints)

Figure 6 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 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 7 shows the optimality rate in five protocols in "cone"-constraints situation.

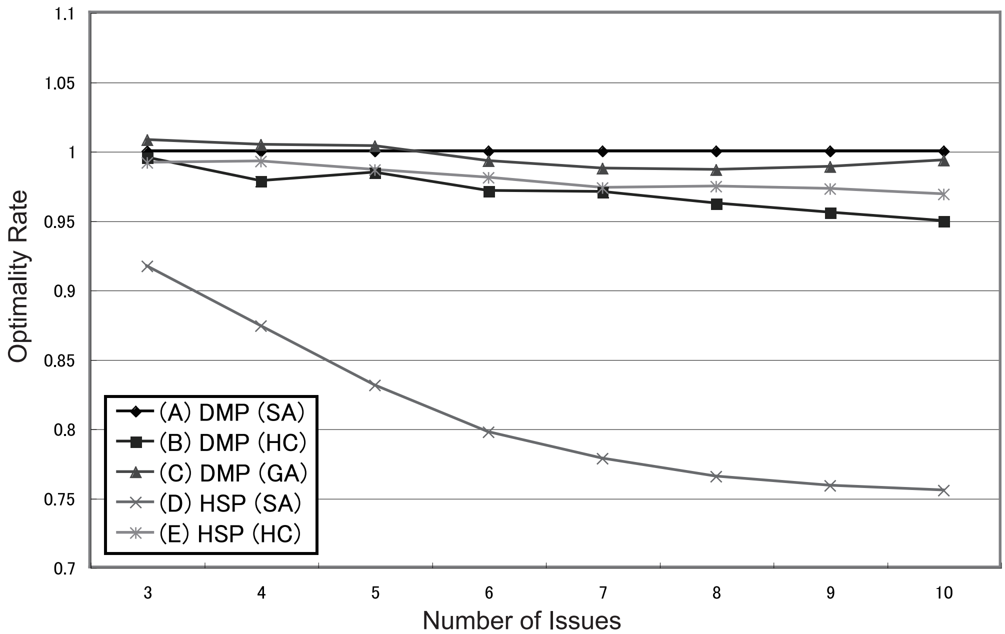

Figure 7: Optimality Rate (Cone-constraints)

"(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 8 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 searches for more states than SA and HC. The number of shares in DMP depends on the features

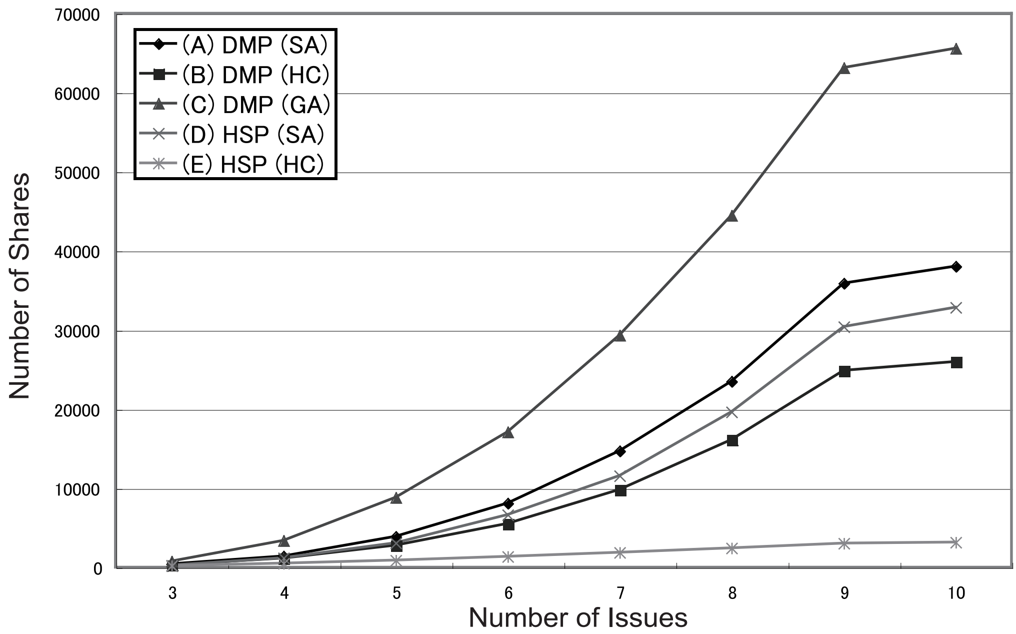

Figure 8: The number of shares (Cube-constraints)

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 8 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

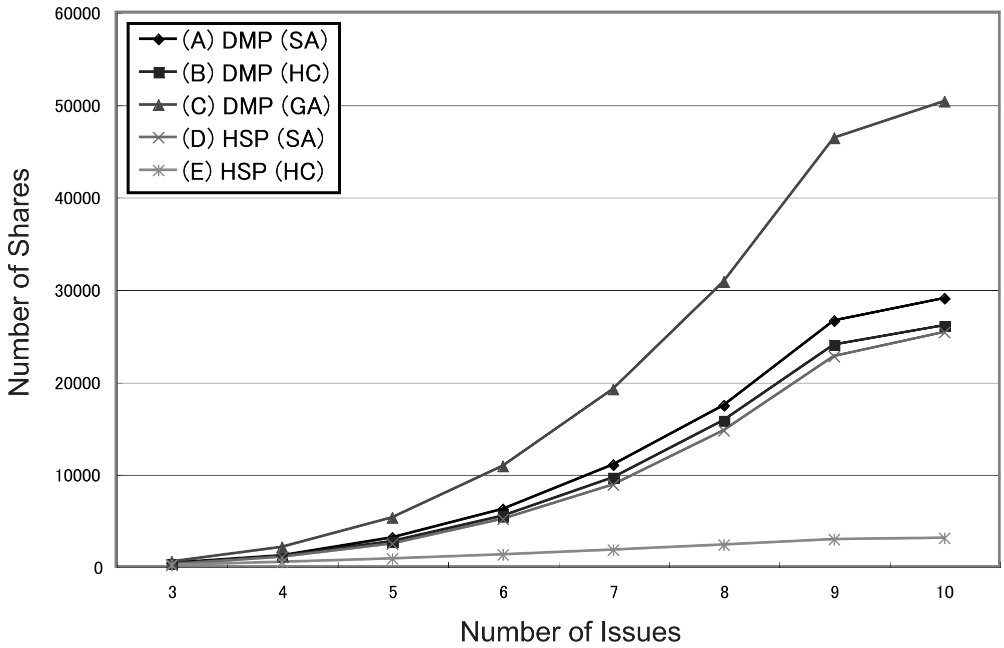

Figure 9: The number of shares (Cone-constraints)

methods in cone-constraints have higher possibility 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) RELATED WORK

Most previous work on multi-issue negotiation [2, 3, 4] has only addressed linear utilities. Recently some researchers have been focusing on more complex and nonlinear utilities.

[15] has explored a range of protocols based on mutation and selection on binary contracts. This paper does not describe what kind of utility function is used, nor does he present any experimental analyses, it is unclear whether this strategy enables

sufficient exploration of utility space.

[1] presents an approach based on constraint relaxation. However, there is no experimental analysis, and this paper merely presents a small toy problem with 27 contracts.

[17] modeled a negotiation problem as a distributed constraint optimization problem. This paper claims the proposed algorithm is optimal, but it does not discuss computational complexity and only provides a single small-scale example.

Based on a simulated-annealing mediator, [12] presented a protocol that was applied with near-optimal results to medium-sized bilateral negotiations with binary dependencies. The work presented here is distinguished by demonstrating both scalability and high optimality values for multilateral negotiations and higher order dependencies.

[13, 14] also presented a protocol for multi-issue problems for bilateral negotiations. [18, 19] presented a multi-item and multi-issue negotiation protocol for bilateral negotiations in electronic commerce situations. [5] proposed bilateral multi-issue negotiations with time constraints, and [22] proposed multi-issue negotiation that employs a third-party to act as a mediator to guide agents toward equitable solutions. This framework also employs an agenda that serves as a schedule for the ordering of issue negotiation. Agendas are very interesting because agents only need to focus on a few issues.

[9] proposed a checking procedure to mitigate this risk and show that by tuning this procedure's parameters, outcome deviation can be controlled. These studies reflect interesting viewpoints, but they focused on just bilateral trading or negotiations.

In previous works ([7] etc.), the utility function has only block constraints, not cone-constraints. The utility function in cone-constraionts is more realistic and complex than these studies. The experiment results in this paper show the performance of some methods in cone-constraints and cube-constraints situation.

(7) CONCLUSION

In this paper, 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.

One future work includes a method to divide the search space depending on agent power. A protocol that develops the scalability of utility information is also possible future work. One possible protocol is to break up the agenda of issues.

REFERENCES

[1] 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.

[2] T. Bosse and C. M. Jonker. Human vs. computer behaviour in multi-issue negotiation. In *Proc. of 1st International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2005)*, pages 11–24, 2005.

[3] 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] 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] 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.

[6] 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] 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.

[8] K. Fujita, T. Ito, and M. Klein. A representative-based multi-round protocol for multi-issue negotiations. In *Proc. of th 7th Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2008)*, 2008.

[9] K. Hindriks, C. Jonker, and D. Tykhonov. Avoiding approximation errors in multi-issue negotiation with issue dependencies. In *Proc. of The 1st International Workshop on Agent-based Complex Automated Negotiations (ACAN-2008)*, 2008.

[10] J. R. V. I. Marsa-Maestre, M. A. Lopez-Carmona 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)*, 2009.

[11] 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.

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

[13] G. Lai, C. Li, and K. Sycara. A general model for pareto optimal multi-attribute negotiations. In *Proc. of The 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)*, 2006.

[14] G. Lai, K. Sycara, and C. Li. A decentralized model for multi-attribute negotiations with incomplete information and general utility functions. In *Proc. of The 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)*, 2006.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] S. J. Russell and P. Norvig. *Artificial Intelligence : A Modern Approach*. Prentice Hall, 2002.

[21] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[22] J. Shew and K. Larson. The blind leading the blind: A third-party model for bilateral multi-issue negotiations under incomplete information. In *Proc. of The 1st International Workshop on Agent-based Complex Automated Negotiations (ACAN-2008)*, 2008.