Representative based Multi-Round Protocol based on Revealed Private Information for Multi-issue Negotiations

Katsuhide Fujita* Takayuki Ito† Mark Klein‡

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 interdependent issues in which agent utility functions are nonlinear. Existing works have not yet focused on the private information of agents. In addition, they were not scalable in the sense that they have shown a high failure rate for making agreements among five or more agents. In this paper, we focus on a novel multi-round representative based protocol that utilizes the amount of revealed agents' private information. Experimental results demonstrate that our mechanism reduces the failure rate in making agreements and is scalable for the number of agents compared with existing approaches.

Keywords: Multi-issue Negotiation, Nonlinear Function

1 Introduction

We envision a future in which large numbers of participants collaborate, negotiate, and reach consensuses through computer-supported negotiation support systems for global problems. Collaboratorium [1, 2], one such pioneering work that enables many people to participate in an argument on such worldwide problems as global warming, provides a platform for coordinating large-scale arguments through web based collaboration tools. In our research, we consider a tool that supports large-scale negotiations among the world's people. In such

*School of Techno-Business Administration Nagoya Institute of Technology/ Visiting Student, Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology, Gokiso, Showa-ku, Nagoya, Aichi, 466-8555 JAPAN, Tel:052-735-7968, Fax:052-735-7404, E-mail:fujita@itolab.mta.nitech.ac.jp

†Visiting Scholar, Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology/ School of Techno-Business Administration, Department of Computer Science and Engineering, Nagoya Institute of Technology/ Researcher, PREST, Japan Science and Technology Agency (JST).

‡Center for Collective Intelligence MIT Sloan School of Management

a situation, eliciting people's utility spaces and automatically finding and suggesting possible agreements would be valuable. People could reach agreements based on such system suggestions.

Multi-issue negotiation protocols represent an important field of study. Even though much previous work exists in this area [3, 4, 5], most of it deals exclusively with simple negotiations involving independent multiple issues. These studies of negotiations mainly assume that agents have an incentive to cooperate to achieve win-win agreements because the situation is not a zero-sum game. Many real-world negotiation problems, however, are complex and involve interdependent multiple issues. Thus, we focus on complex negotiation with interdependent multiple issues.

The bidding based negotiation protocol offers high performance on multi interdependent issues negotiation. However, it has two main issues. **1) Privacy:** Existing works have not yet addressed agents' private information, which should not be revealed excessively because agents who reveal too much utility information suffer a disadvantage. For example, suppose that several companies are collaboratively designing and developing a new car model. If one company reveals more utility information than the other companies, those other companies can learn more of that company's utility information. As a result, the company will face a disadvantage in subsequent negotiations. Furthermore, explicitly revealing utility information is dangerous from a security standpoint. **2) Scalability for the number of agents:** The bidding based negotiation protocol does not have high scalability for the number of agents. In the bidding based negotiation protocol, the mediator needs to find the optimum combination of submitted bids from the agents. However, the computational complexity for finding solutions is too large. The number of agent bids was limited in existing work [6]. Limiting bids causes low optimality and high failure rate for agreements.

To resolve privacy issues, we define an agent's **revealed area** that represents the amount of his/her revealed utility space. This revealed area numerically defines which agents are cooperative and which are not. Additionally, the mediator can understand how much of the agent's private information has been revealed in the negotiation.

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

The representative based protocol has been inspired by the parliamentary systems in England, Canada, Australia, Japan, etc. in which representatives are making an agreement on behalf of other people. In a situation in which many people have to reach an agreement, directly reflecting all members' opinions is quite difficult. Doing so requires much time and energy and is not scalable. Although voting is one option, voting might have paradoxical results [7].

We expand our mechanism to be multi-round by using the Threshold Adjustment Protocol [8]. The multi-round mechanism improves the failure rates and achieves fairness in terms of the revealed area. This means that the amounts of the revealed areas are almost the same among agents. Further, a representative mechanism can prevent the unfair solutions that can exist in the original Threshold Adjustment Protocol.

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

The remainder of this paper is organized as follows. First, we describe a model of non-linear multi-issue negotiation and an existing work's [6] problems. Second, we define the revealed area and proposed our new negotiation mechanism. Third, we describe the multi-round negotiation protocol. Fourth, we present an experimental assessment of this protocol. Finally, we describe related work and draw conclusions.

2 Negotiation Using Complex Utility Space

2.1 Complex Utility Model

We consider the situation where n agents want to reach an agreement. m issues, $s_j \in S$ must be negotiated. The number of issues represents the number of dimensions of the utility space. 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 that has N (e.g., 10) issues. 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.*, $s_j \in [0, X]$. A discrete domain can come arbitrarily close to a real domain by increasing the domain size. As a practical matter, many real-world issues that are theoretically real (delivery date, cost) are discretized during negotiations. Our approach, furthermore, is not theoretically limited to discrete domains. The deal determination part is unaffected, although the bid generation step must be modified to use a nonlinear optimization algorithm suited to real domains. A contract is represented by a vector of issue values $\vec{s} = (s_1, ..., s_m)$.

An agent's utility function is described in terms of constraints. There are

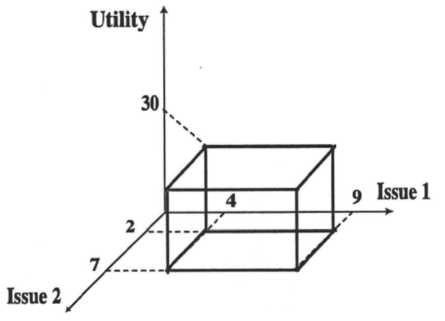

Figure 1: Example of a constraint

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 $\vec{s}$. Figure 1 shows an example of a binary constraint between issues 1 and 2. This constraint has a value of 30 and holds if the value for issue 1 is in the range $[4, 9]$ and the value for issue 2 is in the range $[2, 7]$. Every agent has its own, typically unique, set of constraints.

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. This represents a crucial departure from previous efforts on multi-issue negotiation, where contract utility is calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyperplanes with a single optimum. Figure 2 shows an example of a nonlinear utility space. There are two issues, *i.e.*, two dimensions, with domains $[0, 99]$. There are 50 unary constraints (*i.e.*, that relate to one issue) as well as 100 binary constraints (*i.e.*, that interrelate to two issues). The utility space is highly nonlinear with many hills and valleys.

In our utility function, we assume interdependency between the issues. For example, since an agent has a binary constraint between issues 1 and 2, as Figure 1 shows, they are interdependent for the agent. Therefore, our utility space is highly interdependent.

As is common in negotiation contexts, we assume that agents do not share their utility functions with each other to preserve a competitive edge. In fact, generally agents do not completely know their desired 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 highly uncertain environments.

Finding an optimal contract for individual agents with such utility spaces can be handled using such well-known nonlinear optimization techniques as sim-

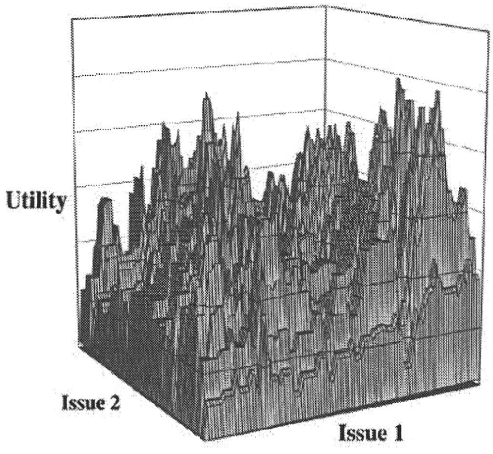

Figure 2: Complex utility space for single agent

ulated annealing or evolutionary algorithms. We cannot employ such methods for negotiation purposes, however, because they require that agents fully reveal their utility functions to a third party, which is generally unrealistic in negotiation contexts.

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

$$\arg\max_{\vec{s}} \sum_{i \in N} u_i(\vec{s}). \tag{1}$$

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

2.2 Existing Bidding based Protocol

In a previous work [6], agents reach an agreement based on the following steps. This is called a **basic bidding based mechanism**.

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

There is no guarantee, of course, that a given sample will lie on a locally optimal contract. Each agent, therefore, uses a nonlinear optimizer based on simulated annealing [9] to find the local optimum in its neighborhood. Figure 3 exemplifies this concept. Black dots are sampling points and white dots are locally optimal contract points.

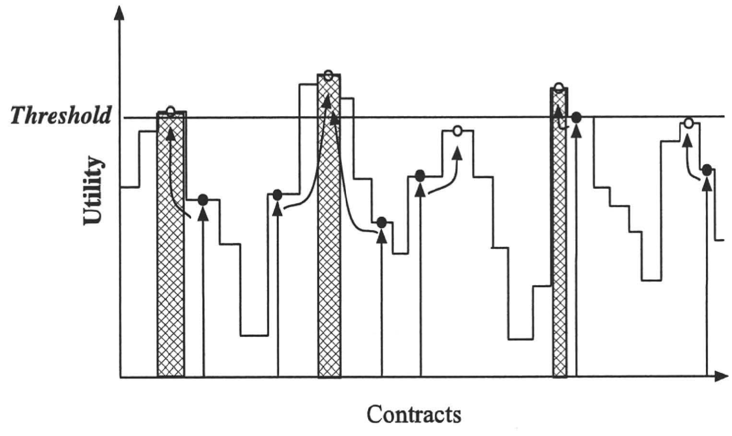

Figure 3: Generating bids

For each contract $\vec{s}$ found by adjusted sampling, an agent evaluates its utility by a summation of the values of satisfied constraints. If that utility is larger than reservation value δ **(threshold)**, then the agent defines a bid that covers all the contracts in the region with that utility value. This is easy: the agent merely finds the intersection of all the constraints satisfied by that $\vec{s}$.

[Find the Solutions] In negotiation, the mediator takes the middle position and 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 (Figure 4) [1]. If there is more than one such overlap, the mediator selects the one with the highest summed bid value (and assuming truthful bidding, the highest social welfare).

2.3 Scalability and Privacy Problems

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

We limited the number of bids for each agent to handle the computational complexity in the basic bidding based protocol [6]. The concrete number of bids in this limitation was $\sqrt[N]{6,400,000}$, a number that reflects our experimental calibration in 2005. But even though CPUs are faster now, the limitation

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

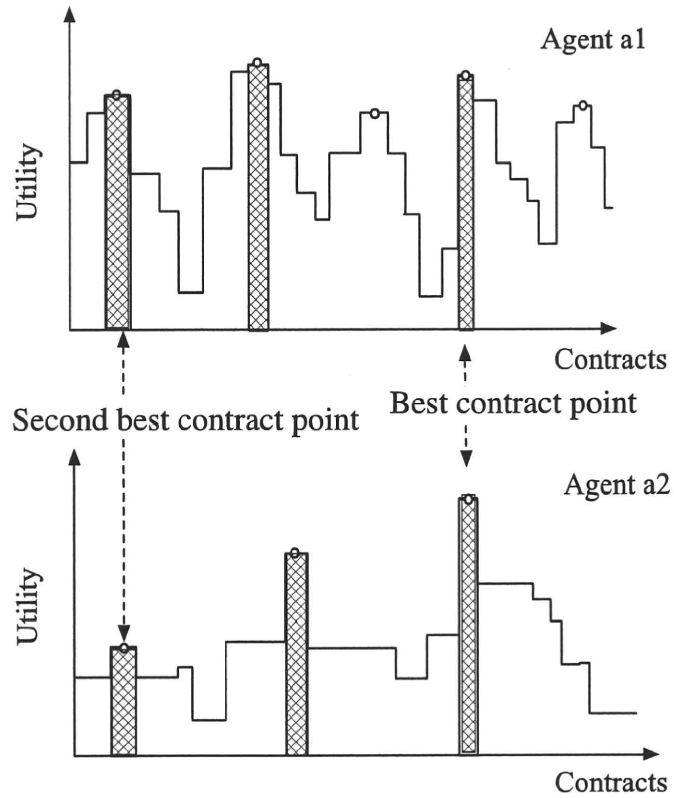

Figure 4: Find solutions

number does not differ so much because this is an exponential problem. Table 1 shows the limitation numbers of bids in one agent. This number quickly drops by increasing the total number of agents. Because of the limitation of bids, the failure rate for finding agreements quickly increases along with increasing the number of agents. When the number of agents is five and the number of issues is seven, we experimentally observed that the failure rate is around 40%. In fact, a strong trade-off exists between increasing the number of total bids and finding good quality solutions. Increasing the number of total bids is not an effective approach for finding good quality agreements.

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

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

Table 1: Limitation of the bids

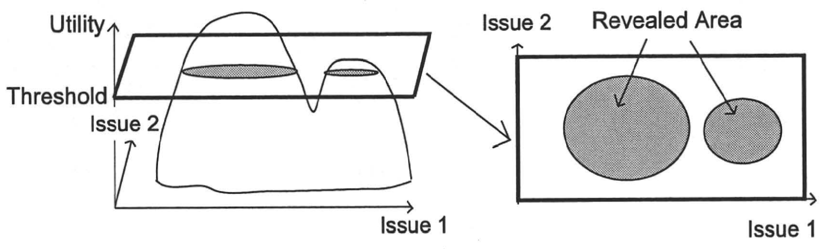

Figure 5: Revealed area

the **revealed area** in utility spaces.

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

3.1 Revealed Area for Agent

We focus on the amount of private information agents revealed in the negotiation. We employ **revealed area** as a measure of the amount of revealed utility space. Figure 5 shows an intuitive example of a revealed area, defined as an agent's possible contract points that are revealed in his utility space on his threshold.

For an agent, it is important for him/her to know how much his/her private information is revealed compared with the other agents. The mediator can judge whether an agent is cooperative based on the amount of revealed private information.

We use a **threshold** that is employed in generating bids as a measure of adjusting agents' revealed areas. Since directly adjusting the revealed area is difficult because agents have complex utility spaces, we consider adjusting their threshold to adjust their revealed areas. The threshold is employed for an agent to generate his/her bids based on utility values above the threshold. The threshold was originally adopted to adjust the number of bids. However, in this paper, we also utilize it for determining an agent's revealed area while handling complex utility space.

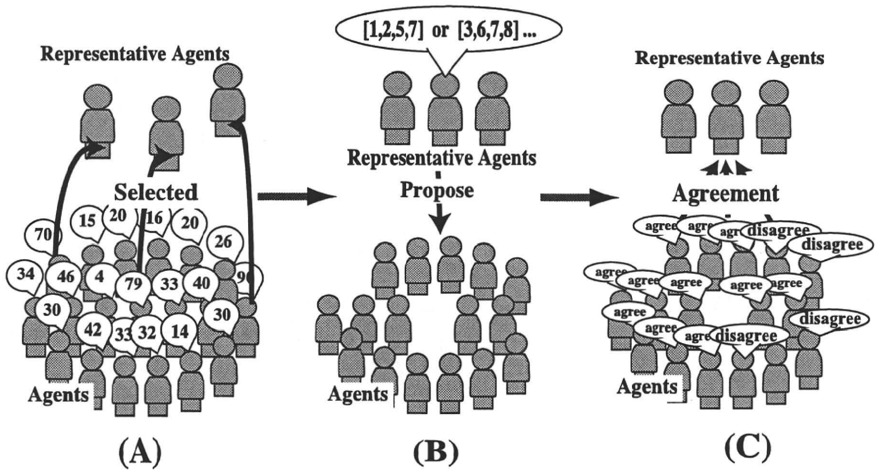

Figure 6: Representative based protocol

3.2 Representative based Protocol

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

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

[Step 1: Selection of Representative Agents] Representative agents are selected based on the amount of their revealed areas, as shown in Figure 6 (A). First, each agent submits how much he can reveal his utility space to the mediator. Namely, each agent submits the numeric value of the amount of his possible revealed area. The mediator selects the representative agents who revealed a large area. We call this selection method RAS. This step is main additional coordination processes by the use of representatives. By employing RAS, all agents are satisfied with the mediator's decision because it is the best method for all agents to find optimal solutions.

[Step 2: Proposing by Representatives] Representative agents find solutions and propose them to other agents, as shown in Figure 6 (B). First, representative agents find solutions by employing a breadth-first search with branch cutting to find solutions (from lines 3 to 14 in representative_protocol()).

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

[Step 3: Respond to Agreement by other Agents] First, the other agents receive the solutions from the representative agents. Then they judge whether they "agree" or "disagree" by determining whether the solution's utility is higher than their own reservation value (Figure 6 (C).).

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

Algorithm 1 representative_protocol(B)

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

$RB := select\_representative(B)$
$SC := RB_0$, $i := 1$
while $i <$ *the number of representative agents* **do**
  $SC' := \emptyset$
  for $s \in SC$ **do**
    for $rb_{i,j} \in RB_i$ **do**
      $s' := s \cup rb_{i,j}$
    end for
  end for
  if s' is consistent **then**
    $SC' := SC' \cup s'$
  end if
  $SC := SC'$, $i := i + 1$
end while
while $i < |SC|$ **do**
  if $ask\_agent(SC_i)$ is true & SC_i Utility is maximum **then**
    return SC_i
  else
    return No Solution
  end if
end while

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

Finally, we describe the trade-off for an agent between revealing a large amount of utility space and being a representative agent. Representative agents have advantages since they can propose alternatives to other agents and dis-

Algorithm 2 ask_agent(SC)

select_representative() is a method for performing Step 1
Th: A reservation value of each agent (Th=$\{Th_0, Th_1, ..., Th_n\}$)

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

advantages because they need to reveal larger utility space. Unrepresentative agents have advantages in keeping their utility hidden and disadvantages in responding based on representatives' agreements.

3.3 Threshold Adjusting Mechanism

We extend our protocol to multi-round negotiation based on the threshold adjusting method [8] so that the number of times to be a representative agent is fair. The total amount of revealed utility space for each agent is almost the same by the threshold adjustment mechanism.

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

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

Figure 7 shows an example of the threshold adjusting process among three agents. The upper and bottom figures show the thresholds and the revealed areas before and after adjusting the threshold, respectively. In particular, in this case, agent 3 revealed a small amount of his utility space. The amount of agent 3's revealed utility space in this threshold adjustment is the largest among these three agents. The exact rate of the amount of revealed utility space and the amount of decreased threshold are defined by the mediator or the mechanism designer.

The threshold adjusting mechanism is shown as Algorithm 3:

In the threshold adjusting mechanism, agents can consider others' behaviors by adjusting the agent thresholds. In our definition, agents can reveal more

Algorithm 3 threshold_adjustment( )

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

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

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

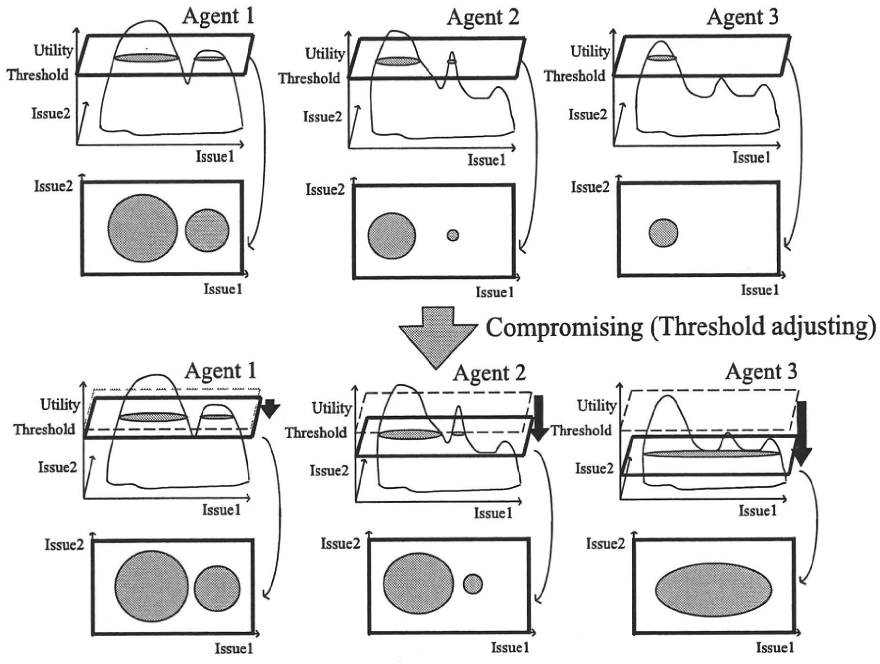

Figure 7: Threshold adjusting process

revealed area if they greatly lower their threshold. Additionally, the width of decreasing the threshold is decided based on a comparison of the others' revealed areas in the threshold adjusting mechanism. Therefore, they can take the behaviors of others into consideration in multi-round negotiation.

4 Experiment Results

4.1 Experiment Settings

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

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

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

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

The following are the parameters for our experiments:

- Domain for issue values: $[0, 9]$.
- Constraints: 10 unary constraints, 5 binary constraints, 5 trinary constraints, etc. (a unary constraint relates to one issue, a binary constraint relates to two issues, and so on).
- Maximum constraint value: $100 \times (number\ of\ issues)$. Constraints that satisfy many issues have on average larger weights. This seems reasonable for many domains. To schedule meetings, for example, higher order constraints concern more people than lower order constraints, so they are more important.
- Maximum constraint width: 7. The following constraints, therefore, are all valid: issue $1 = [2, 6]$, issue $3 = [2, 9]$ and issue $7 = [1, 3]$.
- Number of samples taken during random sampling: $(number\ of\ issues) \times 200$.
- Annealing schedule for sample adjustment: initial temperature 30, 30 iterations. Note that the annealer must not run too long or too 'hot' because then each sample will tend to find the global optimum instead of the peak of the optimum nearest the sampling point.
- Threshold used by agents to select what to bid starts with 900 and decreases until 200 in the threshold adjusting mechanism. The protocol without the threshold adjusting process defines the threshold as 200. The threshold is used to excise contract points with low utility.
- Amount of threshold is decreased by $100 \times (SumAr - Ar_i)/SumAr$. $SumAr$ means the sum of all agents' revealed areas. Ar_i means agent i's revealed area.
- Limitation on number of bids per agent: $\sqrt[n]{6,400,000}$ for N agents. It was only practical to run the deal identification algorithm if it explored no more than about 6,400,000 bid combinations, which implies a limit of $\sqrt[n]{6,400,000}$ bids per agent for N agents.

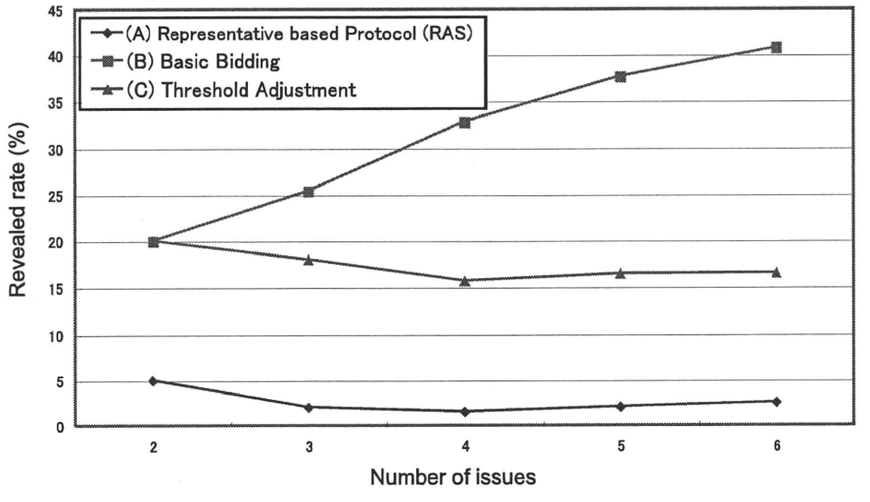

Figure 8: Revealed rate

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

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

4.2 Experimental Results

Figure 8 shows the revealed rate of three comparable protocols with three agents. (A) is the proposed protocol, which is a multi-round negotiation with the representative protocol whose selection method is RAS. (B) is the basic bidding based mechanism without threshold adjustment (explained in Section 2). (C) is the protocol with threshold adjustment.

In (B), the revealed rate increases as the number of issues increases. This means that if we do not use the threshold adjustment, agents need to reveal more of their utility space than the other protocols. On the other hand, in (A) and (C), the revealed rate decreases as the number of issues increases. When we compare (A) with (C), the revealed rate of the representative based protocol is less than basic bidding protocol with threshold adjustment for two reasons. First, the representative protocol finds solutions faster than the threshold adjustment mechanism. Second, bidding based protocol with threshold adjustment most agents need to reveal their utility space. On the other hand, only

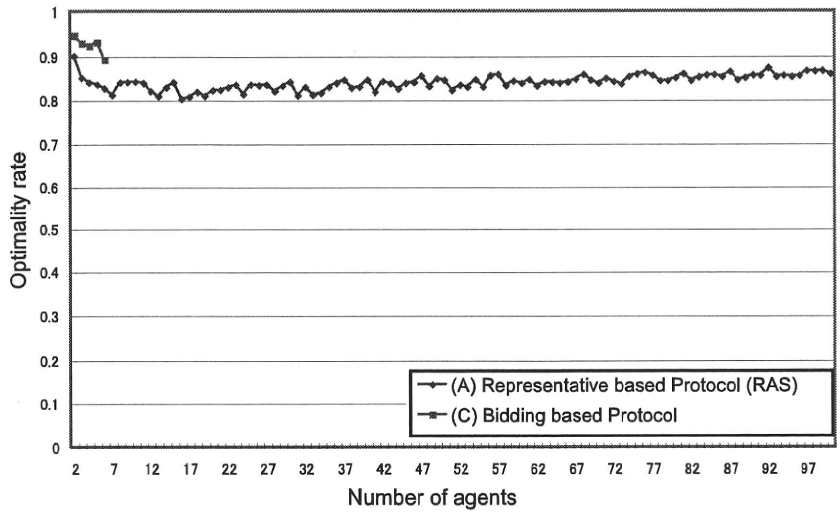

Figure 9: Scalability on number of agents

representative agents reveal their utility spaces in representative based protocol. Essentially, the representative protocol proposed in this paper drastically decreases the revealed rate compared with the other two protocols.

The next experimental results show that our negotiation protocol is sufficiently scalable on the number of agents. Figure 9 shows optimality when the number of agents ranges from 2 to 100. In this experiment, we assume agents have a shared utility area that is agreeable for them. This is because when the number of agents becomes large, finding an agreement point is quite hard using negotiation protocols and comparing optimality could be impossible. To create a common area, agents' utility spaces are randomly generated. Then a common area is randomly generated whose value is more than an agent's threshold.

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

Figure 10 shows the failure rate for finding solutions in the three protocols. (A) is the representative based protocol and selection method RAS. (B) is the representative based protocol and selection method RANDOM. (C) is the basic bidding based mechanism without threshold adjustment explained in Section 2. Even if the number of agents increases, (A) is almost 0. On the other hand, (C) shows a drastic increase over five agents because the bid limitation starts when

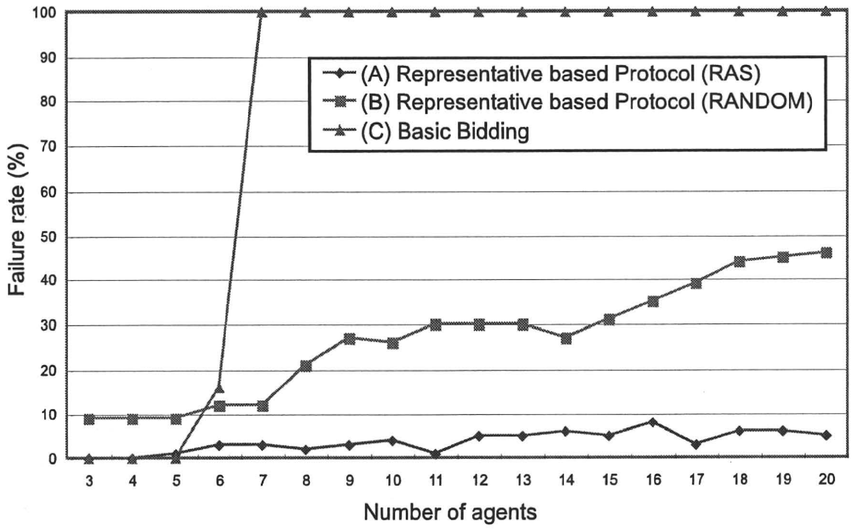

Figure 10: Failure rate

there are five agents. Also, for more than five agents, the existing mechanism fails to find solutions. Furthermore, (A) and (B) show that RAS has a lower failure rate than RANDOM. Thus, the representative protocol with selection method RAS has better failure rates.

Figure 11 compares optimality rates among "(A) Representative based Protocol (RAS)," "(B) Representative based Protocol (RANDOM)," and !H(C) Basic Bidding." Comparing (A) and (C), the difference of optimality is small, around 0.05 at most. This difference reflects that since the representative based protocol tends to find solutions at an early stage, it might miss better solutions. Furthermore, (A) and (B) show that RAS has higher optimality than RANDOM because more solutions are found in representatives with large revealed areas. Thus, the representative protocol with selection method RAS has better optimality rates

Figure 12 shows the variance of the utility value per agent. By this experiment, we can recognize the satisfactory rate of individual agents. The variance of the utility per agent is critical in bargaining theory because some experimental results suggest that fairness influences decision-making per agent ([10] etc.). (C) is better than (A) because the utility of the representatives is higher than that of the unrepresentatives in the representative protocols. However, all agents definitely satisfy the agreement points because their utility values are higher than their reservation value.

Figure 13 shows the optimality and failure rates on the number of representative agents. In this experiment, there are seven agents and three issues.

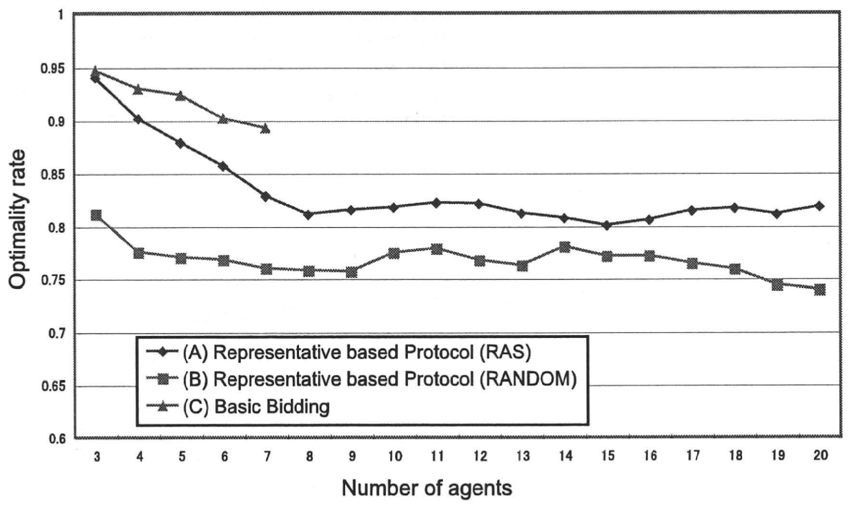

Figure 11: Comparisons of optimality

Figure 13 (A) shows that optimality increases when the number of representative agents increases. Even if agents have low utilities, they tend to be persuaded by representative agents when the number of representatives is small. Figure 13 (B) shows that the failure rate sharply increases when the number of representative agents exceeds, which is where the bid limitation starts.

5 Related Work

Most previous work on multi-issue negotiation [3, 4, 5] has addressed only linear utilities. Recently some researchers have been focusing on more complex and non-linear utilities. [11] does not describe what kind of utility function is used, nor does it present any experimental analyses. It is therefore unclear whether this strategy enables sufficient exploration of the utility space. [12] presents an approach based on constraint relaxation. However, there is no experimental analysis and this paper presents only a small toy problem with 27 contracts. [13] modeled a negotiation problem as a distributed constraint optimization problem. This paper claims the proposed algorithm is optimal, but does not discuss computational complexity and provides only a single small-scale example.

[14] presented a protocol, based on a simulated-annealing mediator, that was applied with near-optimal results to medium-sized bilateral negotiations with binary dependencies. This work is distinguished by demonstrating both scalability and high optimality values for multilateral negotiations and higher order dependencies. [15, 16] also presented a protocol for multi-issue problems for bilateral negotiations. [17, 18] presented a multi-item and multi-issue negotia-

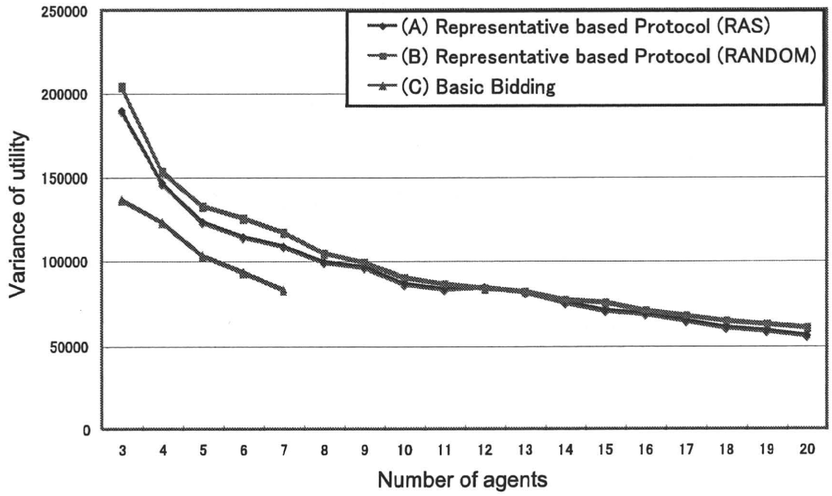

Figure 12: Comparison on variance of utility

tion protocol for bilateral negotiations in electronic commerce situations. [19] proposed bilateral multi-issue negotiations with time constraints. These studies were done from very interesting viewpoints, but focused on just bilateral trading or negotiations.

[20] 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.

[21] 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.

6 Conclusion

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

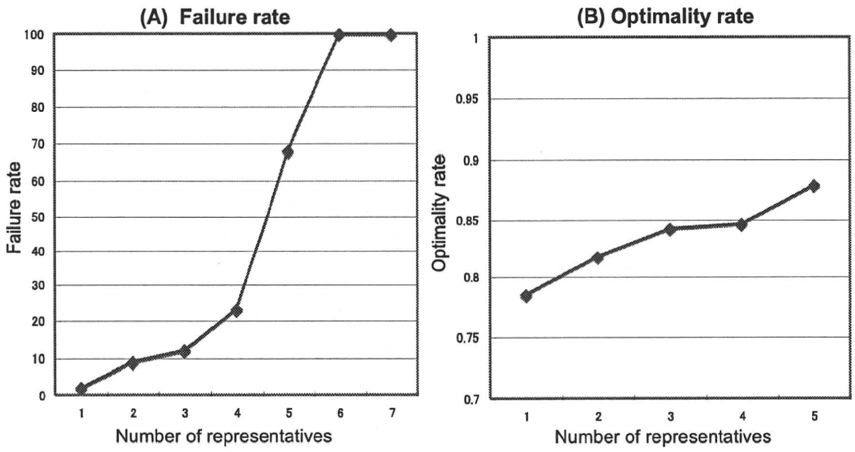

Figure 13: Optimality rate and failure rate in number of representative agents

failure rate in RAS was lower than RANDOM.

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. In the approach described in this paper, 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.

Acknowledgement

This research is partially supported by PREST, JST.

References

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