We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

17

Representative-based Protocol for Multiple Interdependent Issue Negotiation Problems

Katsuhide Fujita
Nagoya Institute of Technology
Japan

Takayuki Ito* and Mark Klein
Massachusetts Institute of Technology
U.S.A.

1. Introduction

Multi-issue negotiation protocols represent an important field in the multi-agent systems study. In fact, negotiation covers many aspects of our lives has led to extensive research in the area of automated negotiators, that is, automated agents capable of negotiating with other agents in a specific environment include e-commerce, large scale argument on worldwide problem(Malone & Klein, 2007), collaborative design for cars and so on. Even though there has been a lot of previous work in this area ((Bosse & Jonker, 2005; Faratin et al., 2002; Fatima et al., 2004; Lin & Chou, 2003)) most have dealt exclusively with simple negotiations involving independent multiple issues. Many real-world negotiations, however, are complex negotiation and involve interdependent issues. When designers work together to design a car, for example, the value of a given carburetor is highly dependent on which engine is chosen. We study on the multiple interdependent issues negotiation using automated agent with nonlinear utility function reflected on the real-world negotiations.
The Bidding-based Negotiation Protocol is high performance on multi interdependent issues negotiation(Ito et al., 2007). In bidding based protocol, agents generate bids by sampling and searching their utility functions, and the mediator finds the optimum combination of submitted bids from agents. However, the bidding-based negotiation protocol has two main issues. **1) Privacy:** Existing works have not yet been concerned with agents' private information. Agents' private information should not be revealed excessively because agents who reveal much utility information are brought to a disadvantage. For example, suppose that several companies collaboratively design and develop a new car model. If one company reveals more utility information than other companies, the other companies can know more of that company's utility information. As a result, the company is brought to a disadvantage in the next negotiations. Furthermore, it is dangerous to reveal utility information explicitly as an aspect of security. **2) Scalability for the number of agents:** The bidding-based negotiation protocol is not a 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 agents. However,

*Visiting from Nagoya Institute of Technology the computational complexity in finding the solution is too large. The number of agent's bids was limited in the exiting work(Ito et al., 2007). Limiting bids have caused low optimality and high failure rate for agreements.

For resolving privacy issues, we define an agent's **revealed area**, which represents the amount of his/her revealed utility space. The revealed area can numerically define 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 the **representative-based protocol** that is high scalability for the number of agents and considering the agent's private information. In our protocol, we first select representatives who revealed their utility space more than the others. These representatives reach an agreement on some alternatives and, propose the alternatives to the other agents. Finally, the other agents can express their own intentions on 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. Namely, although agents tend not to reveal their private information, they have an incentive to reveal their private information in order 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 a lot of people have to make an agreement, it is quite difficult to directly reflect all members' opinions. If we try to do so, it takes much time and energy, and is not scalable. Although voting is one option, it is well known that voting could have paradoxical results (Arrow, 1970).

We expand our mechanism to be multi-round by using the Threshold Adjustment Protocol (Fujita et al., 2007). The multi-round mechanism improves the failure rates and achieve fairness in terms of the revealed area. This means that the amounts of the revealed areas are almost the same among agents. Further, representative mechanism can prevent the unfair solutions that can exist in the original Threshold Adjustment Protocol.

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

The remainder of the paper is organized as follows. First, we describe a model of non-linear multi-issue negotiation and an existing work's(Ito et al., 2007) problems. Second, we define the revealed area and proposed the 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. There are m issues, $s_j \in S$, to be negotiated. The number of issues represents the number of dimensions of the utility space. For example, if there are 3 issues, the utility space has 3 dimensions. The issues are not

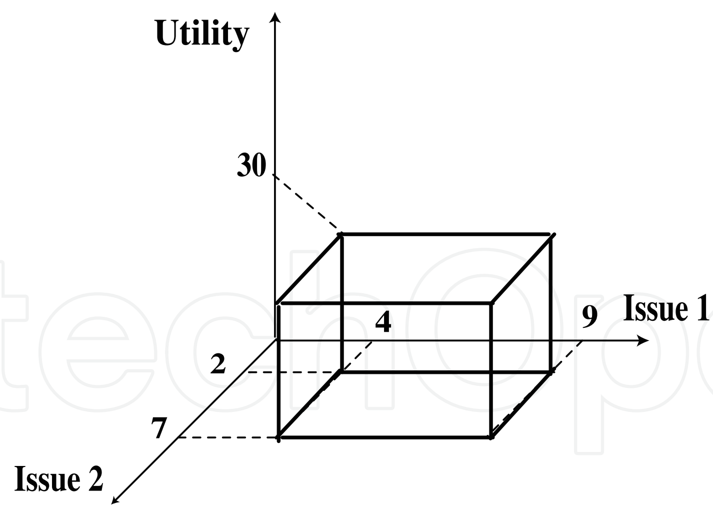


Fig. 1. Example of A Constraint

"distributed" over agents. The agents are all negotiating over a contract that has N (e.g. 10) issues in it. All agents are potentially interested in the values for all N issues. An 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, very 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, though the bid generation step will have to 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 l constraints, $c_k \in C$. Each constraint represents a region with one or more dimensions, and has an associated utility value. A 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 a 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 hyper-planes with a single optimum. Figure 2 shows an example of a nonlinear utility space. There are 2 issues, *i.e.*, 2 dimensions, with domains $[0, 99]$. There are 50 unary constraints (*i.e.*, that relate to 1 issue) as well as 100 binary constraints (*i.e.*, that inter-relate 2 issues). The utility space is, as we can see, highly nonlinear, with many hills and valleys.
We assume, as is common in negotiation contexts, which agents do not share their utility functions with each other, in order to preserve a competitive edge. It will generally be the case, in fact, that agents do not fully know their desirable contracts in advance, because each own utility functions are simply too large. If we have 10 issues with 10 possible values per

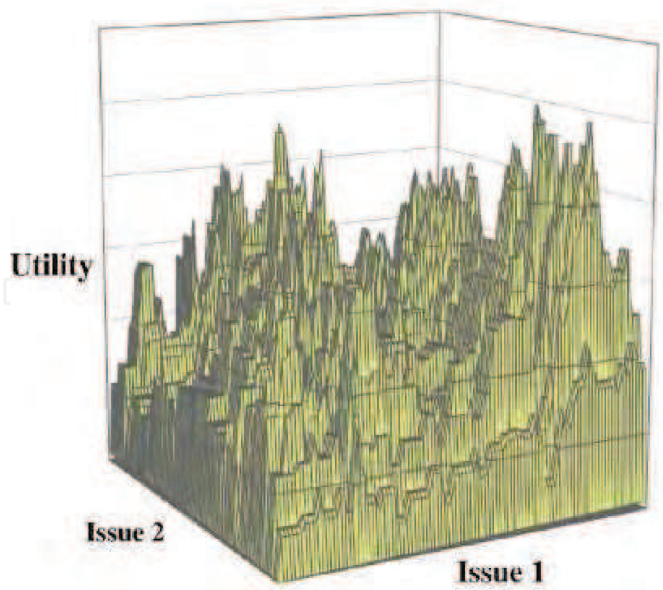


Fig. 2. A Complex Utility Space for a Single Agent

issue, for example, this produces a space of 10^{10} (10 billion) possible contracts, too many to evaluate exhaustively. Agents must thus operate in a highly uncertain environment.
Finding an optimal contract for individual agents with such utility spaces can be handled using well-known nonlinear optimization techniques such a simulated 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:

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

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

2.2 Existing Biding-based Protocol

In the existing work(Ito et al., 2007), agents reach an agreement based on the following steps. We call this **basic bidding-based mechanism**.
[Generate bids] Each agent samples its utility space in order to find high-utility contract regions. A fixed number of samples are taken from a range of random points, drawing from a uniform distribution. Note that, if the number of samples is too low, the agent may miss some high utility regions in its contract space, and thereby potentially end up with a sub-optimal contract.
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 (Russell & Norvig, 2002) to try to find the local optimum in its neighborhood. Figure 3 exemplifies this concept. In this figure, a black dot is a sampling point and a white dot is a locally optimal contract point.

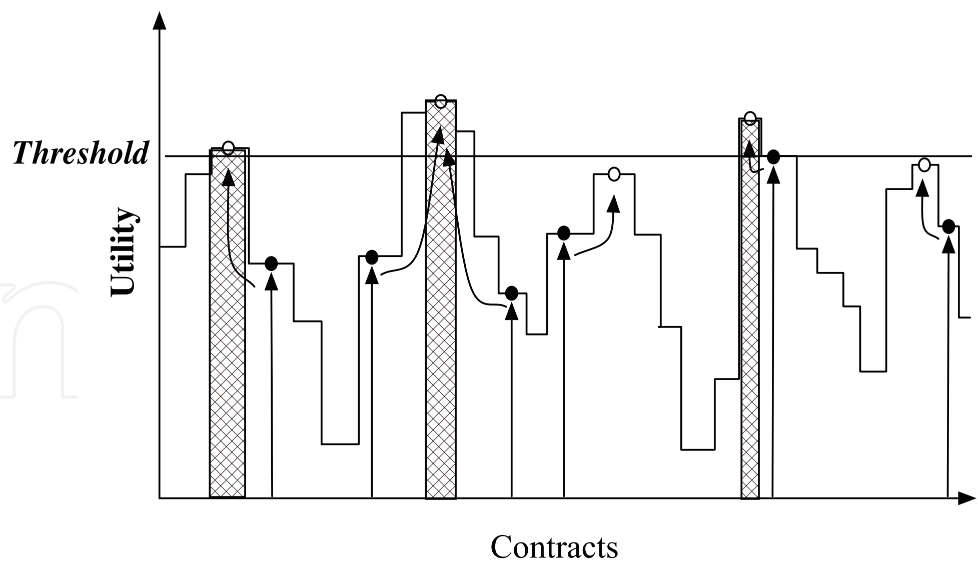


Fig. 3. Example of Generating the Bids

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

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. This is easy to do: the agent need merely find the intersection of all the constraints satisfied by that $\vec{s}$.

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

2.3 Problems on Scalability and Privacy

Computational complexity in finding the solutions exponentially increases according to the number of bids since it is a combinatorial optimization calculation. For example, if there are 10 agents and each agent have 20 bids, the number of bids is 20^{10}. To make our negotiation

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

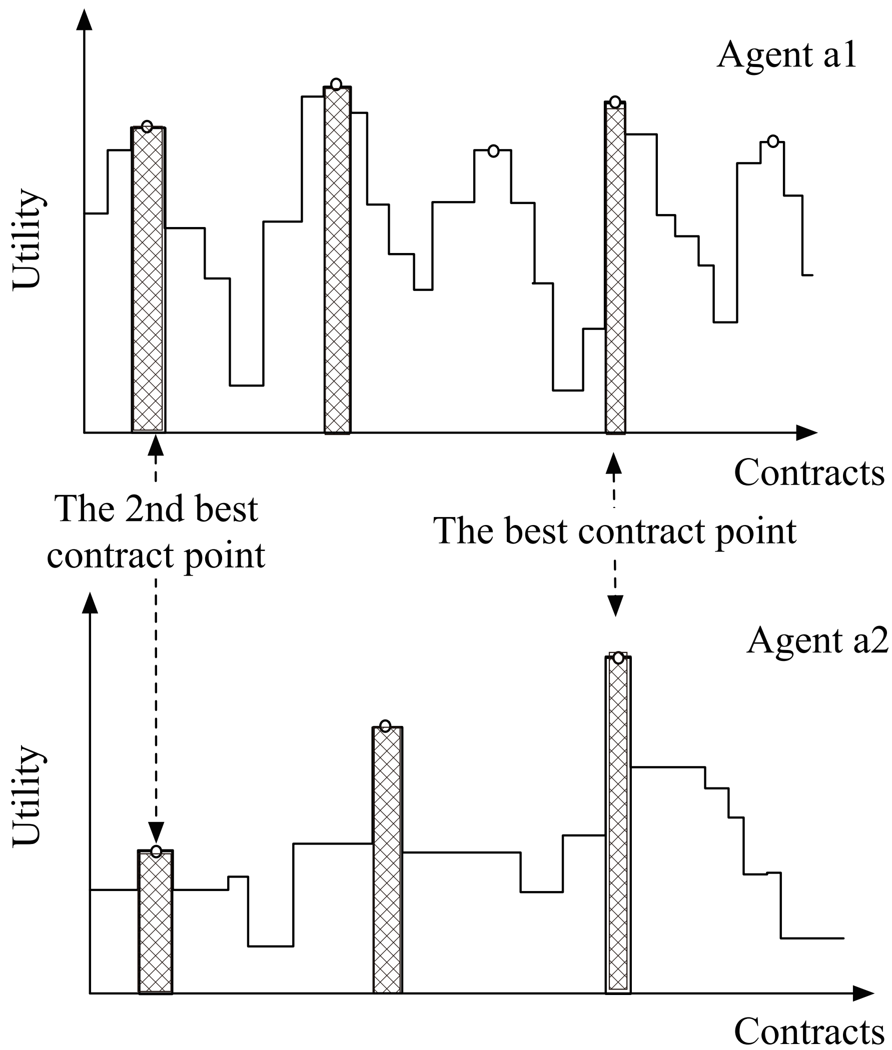


Fig. 4. Find solutions

mechanism scalable, it is necessary to reduce the computational complexity to find the solutions.

In order to handle the computational complexity, in the basic bidding-based protocol (Ito et al., 2007), we limited the number of bids for each agent. The concrete number of bids in this limitation was $\sqrt[N]{6,400,000}$. This number came from our experimental calibration in 2005. But, even though CPUs are faster now, the limitation number does not differ so much because this is an exponential problem. Table 1 shows the limitation numbers of bids in one agent. The limitation number of bids quickly drops by increasing the total number of agents. Because of the limitation of bids, the failure rate in finding agreements quickly increases along with increasing the number of agents. When the number of agents is 5 and the number of issues is 7, we observed experimentally that the failure rate is around 40%. In fact, there is a strong trade-off between just increasing the number of total bids and finding good quality solutions. Thus, increasing the number of total bids is not an effective approach for finding good quality agreements.

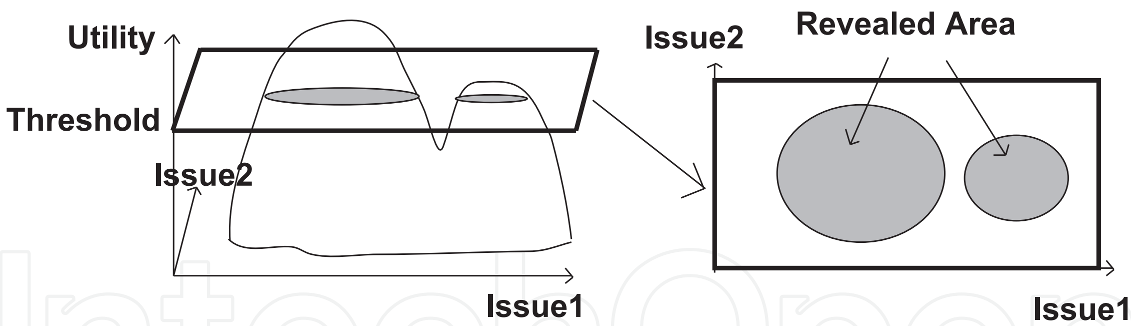


Fig. 5. Revealed Area

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 or security in the utility spaces. Even in a collaborative situation among people, it is normal to keep one's own utility space unopened as long as one is not asked to do otherwise. Our new mechanism will achieve such a situation by defining 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 the measure of the amount of revealed utility space. Figure 5 shows an intuitive example of a revealed area. The revealed area is defined as an agent's possible contract points that are revealed in his utility space on his/her 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 or not cooperative based on his amount of revealed private information.

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

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 for determining whether to "agree" or "disagree" with representative agents. Actually, for practical application, the reservation value can be determined by a human user. Thus, the reservation value is a constant number that is not changed in negotiation. The reservation value is set as lower or the same value as the threshold described in the previous subsection. This protocol consists of following steps.

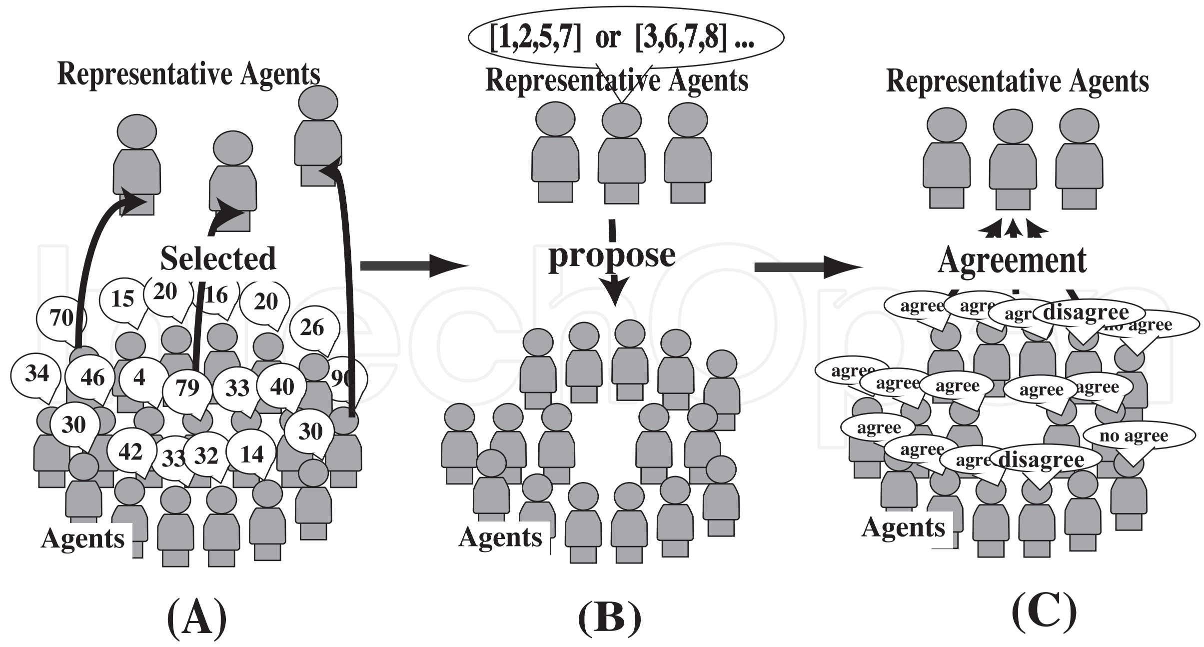


Fig. 6. Representative-based Protocol

[Step1: Selection of the Representative Agents] Representative agents are selected based on the amount of their revealed area as shown in Figure 6 (A). First, each agent submits how much he can reveal his utility 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 could reveal a large area. We call this selection method RAS.

[Step2: Proposing by the Representatives] Representative agents find the solutions and propose to the other agents as shown in Figure 6 (B). First, representative agents find the solutions. They employ a breadth-first search with branch cutting to find solutions. (from line 3 to line 14 in representative_protocol())

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

[Step3: Respond to the agreement by the other agents] The other agents receive the solution from representatives. Each of them will determine whether he/she “agrees” or “disagrees” with the solution (agreement) as shown in Figure 6 (C). First, the other agents receive the solution from the representative agents. Then, they judge whether they will “agree” or “disagree” with the solution. Each agent judges based on whether the solution’s utility is higher than his/her reservation value or not.

Steps 1, 2 and 3 can be captured as Algorithm 12 and Algorithm 2:

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

Finally, we call the trade-off for an agent between revealing a large amount of utility space and being a representative agent. Representative agents have advantages in being able to

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

Algorithm 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
```

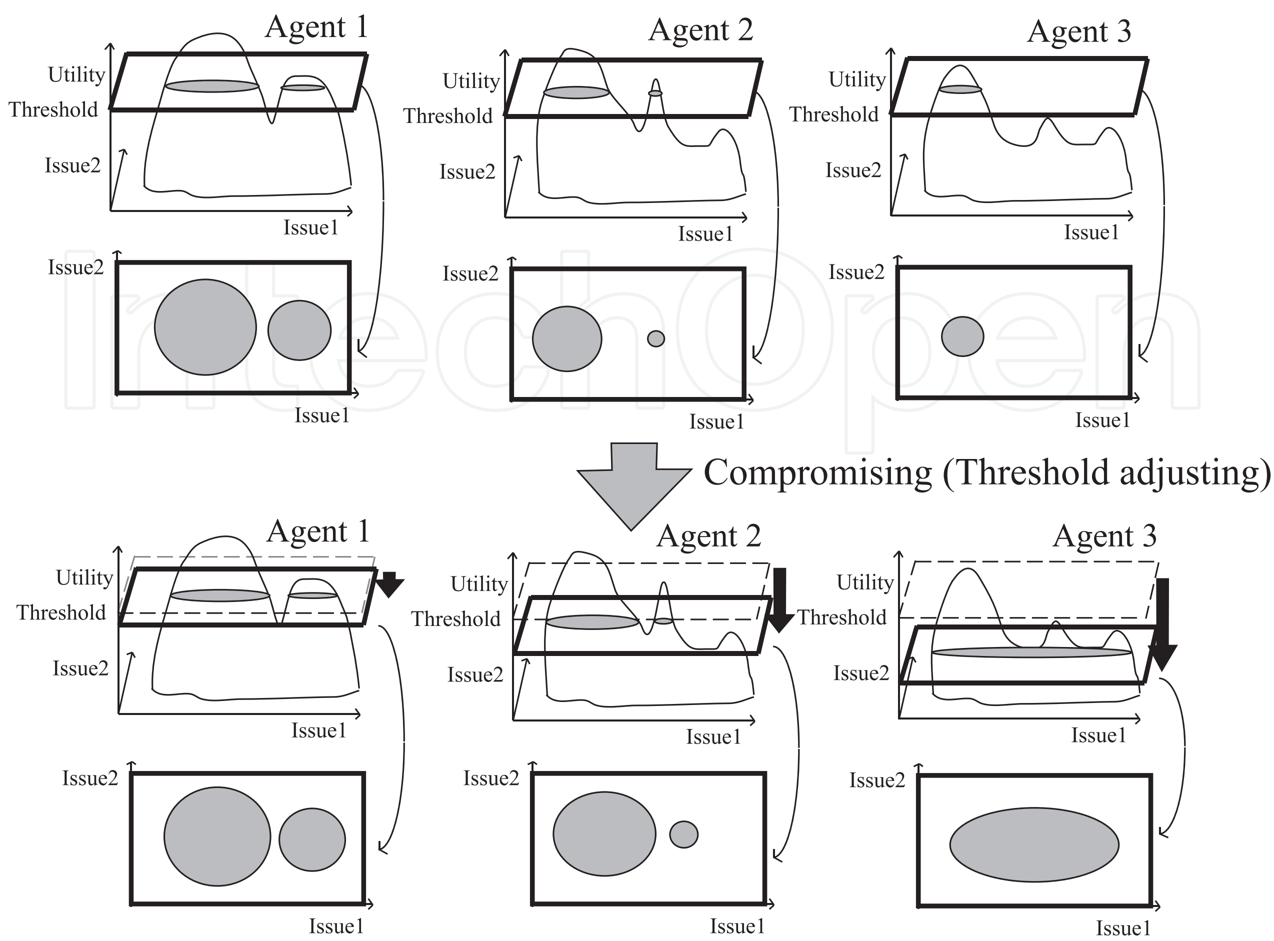


Fig. 7. Threshold Adjusting Process

propose the alternatives to the other agents and disadvantages in the need to reveal larger utility spaces. Unrepresentative agents have advantages in keeping their utility hidden and disadvantages in responding based on the representatives' agreement.

3.3 Threshold Adjusting Mechanism

We extend our protocol to multi-round negotiation based on the threshold adjusting method (Fujita et al., 2007) in order to make the number of times to be representative agents 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 that if an agent reveals a larger area of his utility space, then he should gain an advantage. On the other hand, an agent who reveals a small 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, then this means that he reveals his utility space more.

This mechanism is repeated until an agreement is achieved or all agents refuse to decrease the threshold. Agents can decide whether to decrease the threshold or not based on their reservation value, i.e., the minimum threshold. The reservation value is the limitation that of what the agent can reveal. This means that agents have the right to reject the request to decrease their threshold if the request decreases the threshold lower than the reservation value.

Figure 7 shows an example of the threshold adjusting process among 3 agents. The upper figure shows the thresholds and the revealed areas before adjusting the threshold. The bottom figure shows the thresholds and the revealed areas after adjusting the threshold. In particular, in this case, agent 3 revealed 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 3 agents. The exact rate of the amount of utility space revealed and the amount of threshold decreased is defined by the mediator or the mechanism designer.
The threshold adjusting mechanism is shown as Algorithm 3:

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 := ∅
   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 := Σ_{i∈|Ag|} Ar_i
      while i < |Ag| do
         Th_i := Th_i − C ∗ (SumAr − Ar_i)/SumAr
         i := i + 1
      end while
   else
      break loop
   end if
end loop
return maxSolution
```

4. Experiment Results

4.1 Setting of Experiment

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 measure is the range of revealed area. Namely, if an agent reveals one point on the grid of utility space, this means he lost 1 privacy unit. If he reveals 1000 points, then he lost 1000 privacy units.
We also analyze the representative selection method in our protocol. The representative selection method has remained an important research point. We call the selection method in which agents who reveal a larger utility area are selected representatives (**RAS**), and the random selection method in which representatives are randomly selected (**RANDOM**). To investigate the detailed effects of RAS, we assume RANDOM is the general basis for comparison.
The parameters for our experiments were as follows:

- Domain for issue values is $[0, 9]$.
- 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).
- The maximum value for a constraint: $100 \times (Number\ of\ Issues)$. Constraints that satisfy many issues thus have, on average, larger weights. This seems reasonable for many domains. In meeting scheduling, for example, higher order constraints concern more people than lower order constraints, so they are more important for that reason.
- The maximum width for a constraint: 7. The following constraints, therefore, would all be valid: issue 1 = $[2, 6]$, issue 3 = $[2, 9]$ and issue 7 = $[1, 3]$.
- The number of samples taken during random sampling: $(Number\ of\ Issues) \times 200$.
- Annealing schedule for sample adjustment: initial temperature 30, 30 iterations. Note that it is important that the annealer not run too long or too 'hot' because then each sample will tend to find the global optimum instead of the peak of the optimum nearest the sampling point.
- The threshold agents used to select which bids to make start with 900 and decrease until 200 in the threshold adjusting mechanism. The protocol without the threshold adjusting process defines the threshold as 200. The threshold is used to cut out contract points that have low utility.
- The amount of the threshold is decreased by $100 \times (SumAr - Ar_i)/SumAr$. $SumAr$ means the sum of all agents' revealed area. Ar_i means agent i's revealed area.
- The limitation on the 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.
- The number of representative agents is 2 in the representative-bsed protocol.
- The number of issues is 3.

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 the Mac OS X 10.4 operating system.

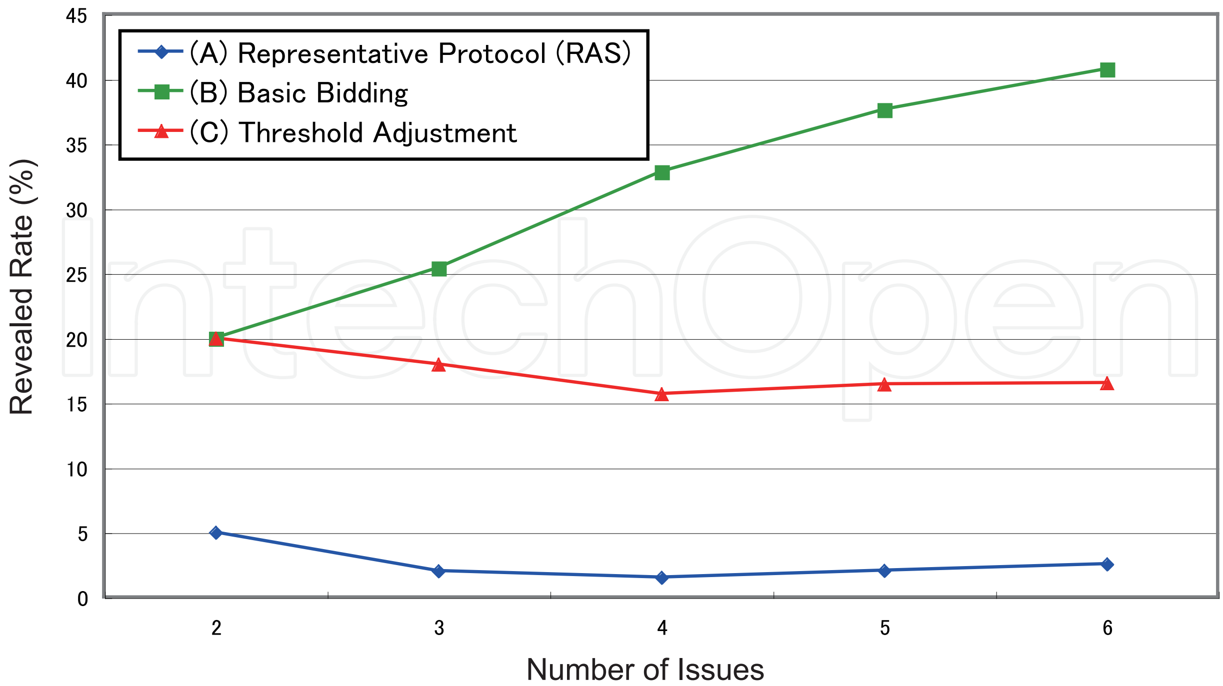


Fig. 8. Revealed Rate

4.2 Experimental Results

Figure 8 shows the revealed rate of 3 comparable protocols. The number of agents is 3. (A) is the proposed protocol that is a multi-round negotiation with the representative protocol the 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 reveal their utility space too much more 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 protocol is less than the mechanism with threshold adjustment. There are two reasons for this. First, the representative protocol finds the solutions earlier than the threshold adjustment mechanism. Second, in the threshold adjustment most agents need to reveal their utility space. On the other hand, only representative agents reveal their utility spaces. Essentially, the representative protocol proposed in this paper drastically decreases the revealed rate compared with the other two protocols.

The next experimental results show our negotiation protocol is sufficiently scalable on the number of agents. Figure 9 shows the optimality when agents reach an agreement when the number of agents is 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, it is quite hard to find an agreement point by using any negotiation protocols and it could be impossible to compare optimality. To create a common area, first, agents' utility space is randomly generated. Then, a common area whose value is more than an agent's threshold is randomly generated. The results demonstrated that the optimality is more than 80% in all cases. Although the high optimality came from the above common area assumption, scalability of our new protocol is ensured by this experiment.

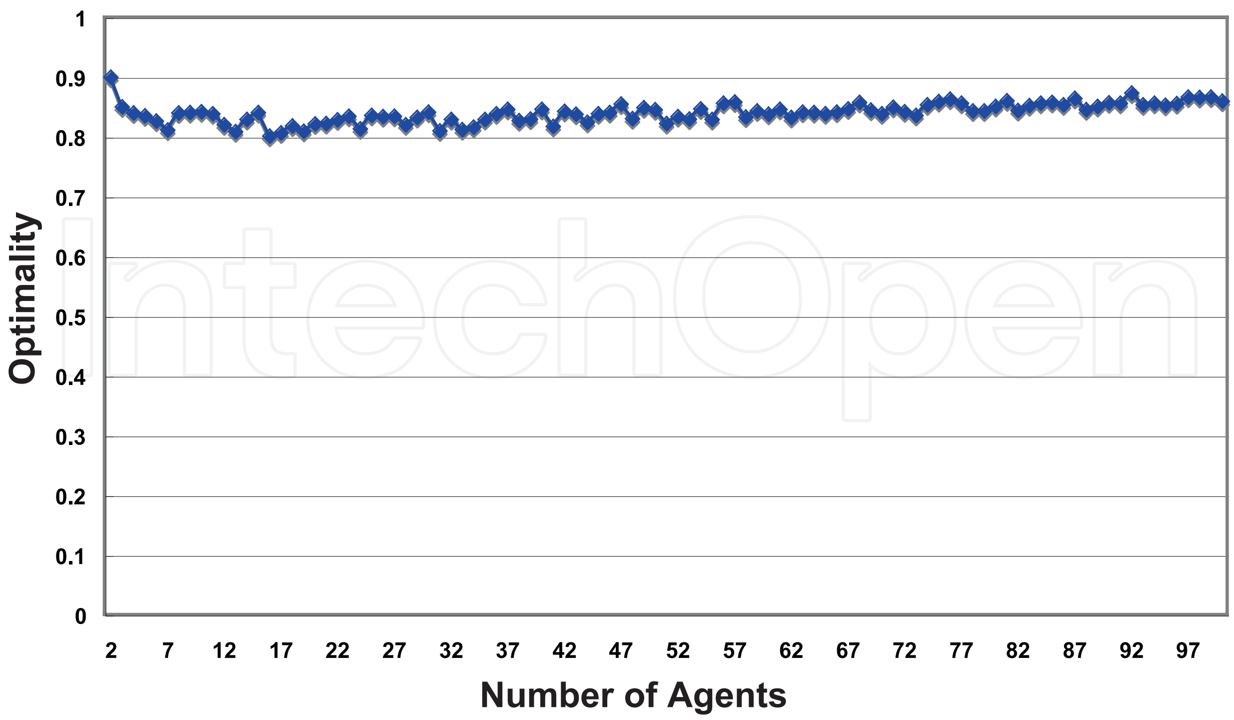


Fig. 9. Scalability on number of agents

Figure 10 shows the failure rate in finding solutions in the three protocols. (A) is the proposed protocol that is a multi-round negotiation with the representative protocol the selection method is RAS. (B) is the representative-based protocol with the representative protocol the selection method is 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 5 agents. This is because the bid limitation for computing winner determination starts when there are 5 agents. Also, for more than 5 agents, the existing mechanism fails to find solutions. Furthermore, (A) and (B) show that RAS improves the failure rate more than RANDOM. Thus, the representative protocol with the selection method is RAS has better failure rates.

Figure 11 shows a comparison on optimality rate among (A), (B) and (C). Comparing (A) and (C), the difference of optimality is small, and around 0.05 at most. This difference comes from the fact that since the representative-based protocol tends to find the solutions at an early stage, it might miss better solutions. Furthermore, (A) and (B) show that RAS is higher optimality than RANDOM. The reason for this is that more solutions are found in representatives who have large revealed area. Thus, the representative protocol with the selection method is RAS has better optimality rates.

Figure 12 shows the variance of the number of times to be representative agents in (A) and (D). The fairness of the number of times to be representative agents is defined as the variance of the number of times to be representative agents for each agent. Comparing (A) with (D), the deviation of the (A) is much lower than that of (D). Thus, RAS can achieve fair opportunity on the number of times to be representative agents.

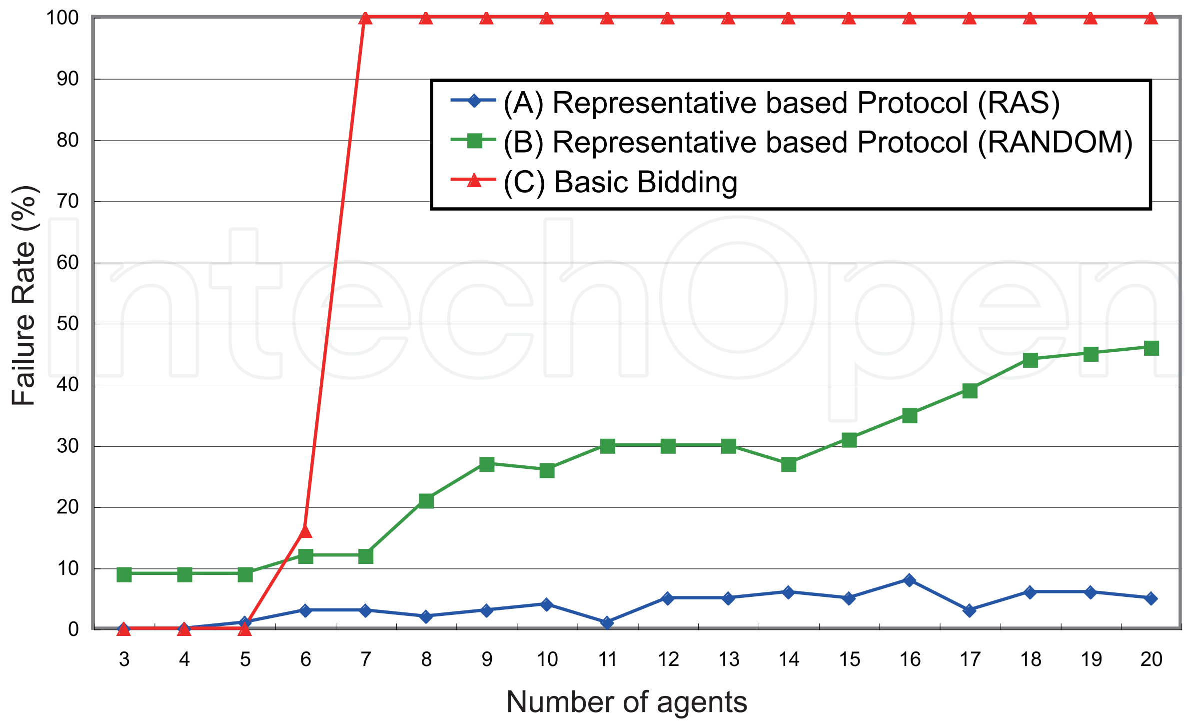


Fig. 10. Failure Rate

5. Related Work

Most previous work on multi-issue negotiation (Bosse & Jonker, 2005; Faratin et al., 2002; Fatima et al., 2004) has addressed only linear utilities. Recently some researchers have been focusing on more complex and non-linear utilities. (Lin & Chou, 2003) 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. (Barbuceanu & Lo, 2000) 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. (Luo et al., 2003) 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.

(Klein et al., 2003) presented a protocol, based on a simulated-annealing mediator, 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. (Lai, Li & Sycara, 2006; Lai, Sycara & Li, 2006) also presented a protocol for multi-issue problems for bilateral negotiations. (Robu & Poutre, 2006; Robu et al., 2005) presented a multi-item and multi-issue negotiation protocol for bilateral negotiations in electronic commerce situations. (Fatima et al., 2007) proposed bilateral multi-issue negotiations with time constraints. These studies were done from very interesting viewpoints, but focused on just bilateral trading or negotiations.

(Shew & Larson, 2008) 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

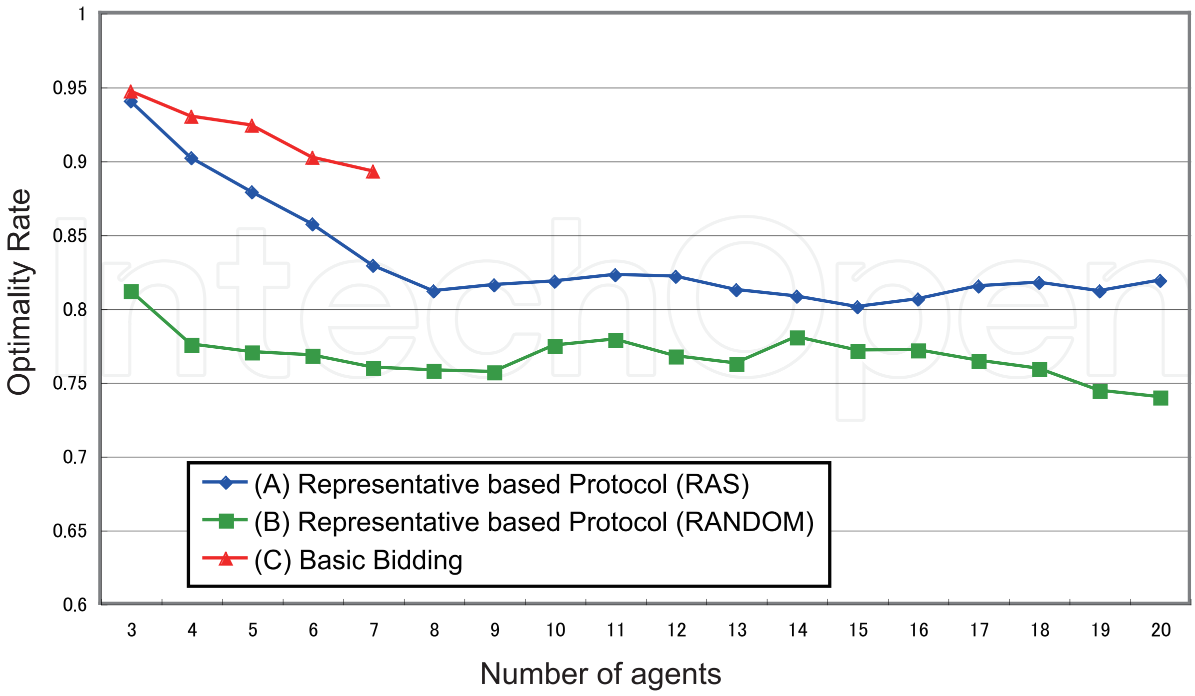


Fig. 11. Comparison on Optimality

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.
(Hindriks et al., 2008) 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 could always make agreements if the number of agents was large. It was important for agents to make agreements without revealing their private information in the negotiation. This proposed protocol could reach an agreement while revealing agents' utility space as little as possible. The experimental results demonstrated that the representative-based protocol could reduce the amount of private information that is required for an agreement among agents, and the failure rate in this mechanism was almost 0. Furthermore, we compared RAS with RANDOM in the experiments. The failure rate in RAS was lower than RANDOM.
In terms of possible future work, 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 the paper, the representative's utility functions are purely idiosyncratic to them, so the solutions that the representatives like may be different from the solutions that are best for the other agents. Changing representatives in the mluti round negotiation helps to support this. The changing mechanism proposed here is a simple one. Thus investigating changing mechanisms are possible future work.

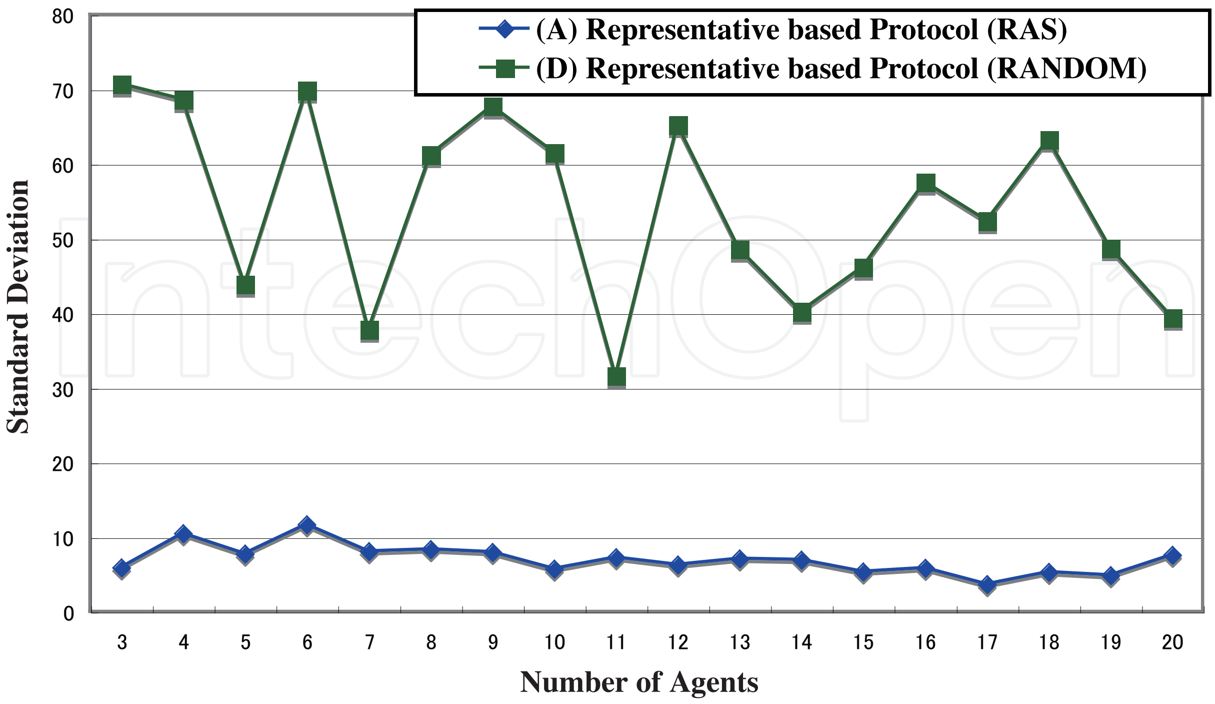


Fig. 12. Variance of the number of times to be representative agents

7. References

Arrow, K. J. (1970). *Social Choice and Individual Values*, Yale Univ. Press.

Barbuceanu, M. & Lo, W.-K. (2000). Multi-attribute utility theoretic negotiation for electronic commerce, *In Proceedings of the International Workshop on Agent-mediated Electronic Commerce (AMEC2000)*, pp. 15–30.

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

Faratin, P., Sierra, C. & Jennings, N. R. (2002). Using similarity criteria to make issue trade-offs in automated negotiations, *Artificial Intelligence*, pp. 142:205–237.

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

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

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

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

Ito, T., Hattori, H. & Klein, M. (2007). Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces, *In Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pp. 1347–1352.

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

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

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

Lin, R. J. & Chou, S. T. (2003). Bilateral multi-issue negotiations in a dynamic environment, *In Proceedings of the International Workshop on Agent-mediated Electronic Commerce (AMEC2003)*.

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

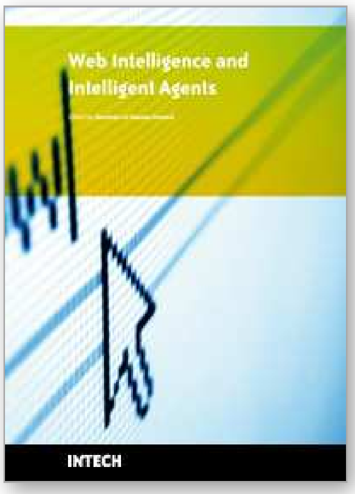
Malone, T. W. & Klein, M. (2007). Harnessing collective intelligence to address global climate change, *Innovations Journal* **2(3)**: 15–26.

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

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

Russell, S. J. & Norvig, P. (2002). *Artificial Intelligence : A Modern Approach*, Prentice Hall.

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



Web Intelligence and Intelligent Agents
Edited by Zeeshan-Ul-Hassan Usmani

ISBN 978-953-7619-85-5
Hard cover, 486 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents a unique and diversified collection of research work ranging from controlling the activities in virtual world to optimization of productivity in games, from collaborative recommendations to populate an open computational environment with autonomous hypothetical reasoning, and from dynamic health portal to measuring information quality, correctness, and readability from the web.