An Analysis of Computational Complexity of the Threshold Adjusting Mechanism in Multi-Issue Negotiations

Katsuhide Fujita and Takayuki Ito

Dept. of Computer Science Nagoya Institute of Technology, Nagoya, JAPAN
Techno-Business School / Dept. of Computer Science, Nagoya Institute of Technology, Nagoya, JAPAN
Email: fujita@itolab.mta.nitech.ac.jp, ito.takayuki@nitech.ac.jp

Abstract: In this paper, we propose a threshold adjusting mechanism in complex negotiations among software agents. The mechanism reduces computational complexity to find agreements that produce higher social welfare. Multi-issue negotiation protocols have been studied widely and represent a promising field since most negotiation problems in the real world involve interdependent multiple issues. Our work focuses on negotiation with interdependent issues, in which agent utility functions are nonlinear. We have proposed negotiation protocols where a bidding-based mechanism is used to find social-welfare maximizing deals. The existing works have not succeeded on reducing the computational cost for finding an agreement. The threshold adjusting mechanism proposed here reduces the number of candidate bids. The experimental results show that the threshold adjusting mechanism can reduce the computational cost while keeping enough optimality. Additionally, we demonstrate threshold adjusting mechanism can achieve the fairness of the revealing the each agent's private information.

1. Introduction

Multi-issue negotiation protocols represent an important field of study since negotiation problems in the real world are often complex ones involving multiple issues. While there has been a lot of previous work in this area ([3, 4, 14, 21]) these efforts have, to date, dealt almost exclusively with simple negotiations involving **independent multiple** issues, and therefore linear (single optimum) utility functions. Many real-world negotiation problems, however, involve **interdependent multiple** issues. When designers work together to design a car, for example, the value of a given carburetor is highly dependent on which engine is chosen. The addition of such interdependencies greatly complicates the agent's utility functions, making them nonlinear, with multiple optima. Negotiation mechanisms that are well suited for linear utility functions, unfortunately, fare poorly when applied to nonlinear problems ([11]).

We have proposed a bidding-based **multiple-issue** negotiation protocol suited for agents with such nonlinear utility functions[7, 8, 9, 10]. Agents generate bids by sampling their own utility functions to find local optima, and then using constraint-based bids to compactly describe regions that have large utility values for that agent. These techniques make bid generation computationally tractable even in large (*e.g.*, 10^{10} contracts) utility spaces. A mediator then finds a combination of bids that maximizes social welfare.

The existing works have not yet concerned about agents' private information. Such private information should be kept as much as possible in their negotiation. If all agents' utility is revealed, other agents can know their private information. As a result, the agents are brought to a disadvantage in the next negotiations. Furthermore, it is dangerous to reveal utility information explicitly as an aspect of security.

For example, suppose that more than 2 companies work together to design a car. The value of a given carburetor highly depends on which engine is chosen. A company will have a disadvantage in the next negotiation if they reveal their entire utility space. The goal is to decide the optimum car design without revealing their entire utility space. We propose the automated negotiation mechanism to make agreements like in such negotiation problems.

Also, there has been a strong limitation on the computational cost for finding the best deal. 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 mechanism scalable, it is necessary to reduce the computational complexity to find the solutions.

In this paper, 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 a threshold adjusting mechanism. First agents make bids that produce more utility than the common threshold value based on the bidding-based protocol we proposed in [7, 8, 9, 10]. Then the mediator asks each agent to reduce its threshold based on how much each agent opens its private information to the others. Each agent makes bids again above the threshold. This process continues iteratively until agreement is reached or no solution.

This mechanism has the following two features: (1) The mechanism reduces the computational cost for finding agreements that can produce higher social welfare. (2) The mechanism can facilitate agents to reach an agreement while keeping their private information as much as possible. We have discussed feature (2) in the previous report [6]. This paper focuses on feature (1). Our experimental results show that our method substantially outperforms the existing negotiation methods on the point of the computational costs. Additionally, we demonstrate threshold adjusting mechanism can achieve the fairness of the revealing the each agent's private information.

The remainder of the paper is organized as follows. First we describe a model of non-linear multi-issue negotiation. Second, we describe the outline of a bidding-based negotia-

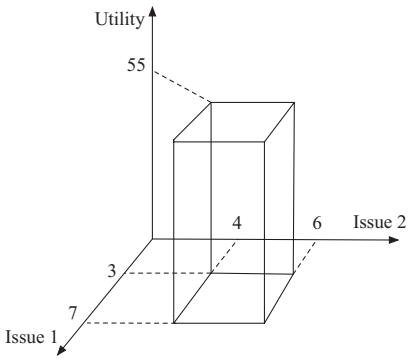


Fig. 1. Example of A Constraint

tion protocol designed for such contexts. Third we propose a threshold adjusting mechanism that reduces the computational cost. Forth, we present experimental assessment of this protocol. Finally, we compare our work with previous efforts, and conclude with a discussion of possible avenues for future work.

2. Complex Negotiation

2.1 Complex Utility Space

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[1], the utility space has 3 dimensions. An issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in [0, X]$[2].

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, $c_k \in C$ (C: a set of constraints). That is to say, the agent has l constraints, and we describe kth constraint as c_k. 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 55, and holds if the value for issue 1 is in the range $[3, 7]$ and the value for issue 2 is in the range $[4, 6]$. 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 hyperplanes 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.

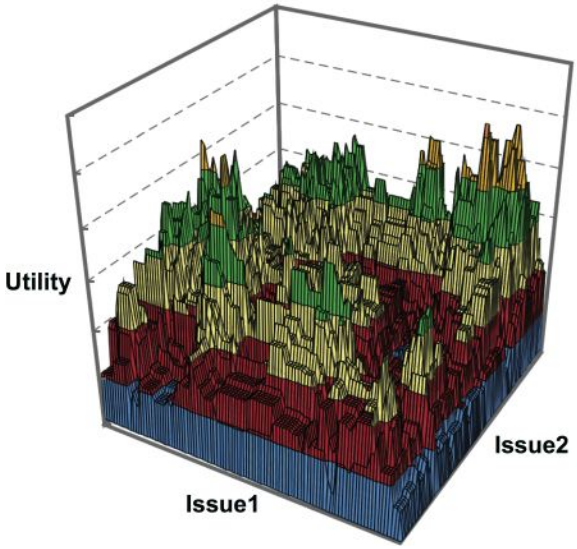


Fig. 2. An Utility Space for a Single Agent

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

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

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

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 Bidding-based Consenting Protocol

The bidding-based negotiation protocol[8] consists of the following four steps:

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

[1] The issues are not "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.

[2] 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.

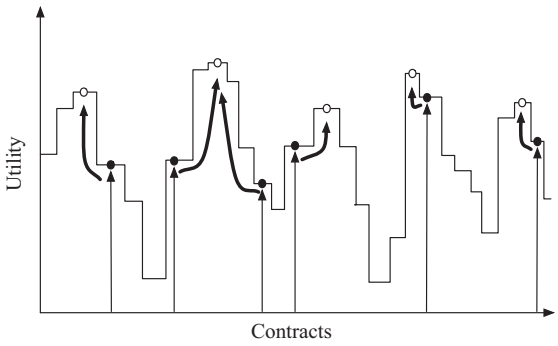


Fig. 3. Adjusting the Sampled Contract Points

[Step 2: Adjusting]
There is no guarantee, of course, that a given sample will lie on a locally optimal contract. Each agent, therefore, uses a nonlinear optimizer based on simulated annealing to try to find the local optimum in its neighborhood. Figure 3 exemplifies this concept. In this figure, a black dot is a sampling point and a white dot is a locally optimal contract point.

[Step 3: Bidding]
For each contract $\vec{s}$ found by adjusted sampling, an agent evaluates its utility by summation of values of satisfied constraints. If that utility is larger than the reservation value δ, then the agent defines a bid that covers all the contracts in the region that has that utility value. This is easy to do: the agent need merely find the intersection of all the constraints satisfied by that $\vec{s}$.

Steps 1, 2 and 3 can be captured as follows:
SN: The number of samples
T: Temperature for Simulated Annealing
V: A set of values for each issue, V_m is for an issue m

procedure bid-generation with SA(*Th*, *SN*, *V*, *T*, *B*)
$P_{smpl} := \emptyset$
while $|P_{smpl}| < SN$
  $P_{smpl} := P_{smpl} \cup \{p_i\}$ (randomly selected from P)
$P := \Pi_{m=0}^{|I|} V_m$, $P_{sa} := \emptyset$
for each $p \in P_{smpl}$ **do**
  $p' :=$ simulatedAnnealing(p, T)
  $P_{sa} := P_{sa} \cup \{p'\}$
for each $p \in P_{sa}$ **do**
  $u := 0$, $B := \emptyset$, $BC := \emptyset$
  for each $c \in C$ **do**
    if c contains p as a contract and p satisfies c **then**
      $BC := BC \cup c$
      $u := u + v_c$
  if $u >= Th$ **then**
    $B := B \cup (u, BC)$

[Step 4: Deal identification]
The mediator identifies the final contract by finding all the combinations of bids, one from each agent, that are mutually consistent, *i.e.*, that specify overlapping contract regions. If there is more than one such overlap, the mediator selects the one with the highest summed bid value (and thus, assuming truthful bidding, the highest social welfare)(see Figure 4).

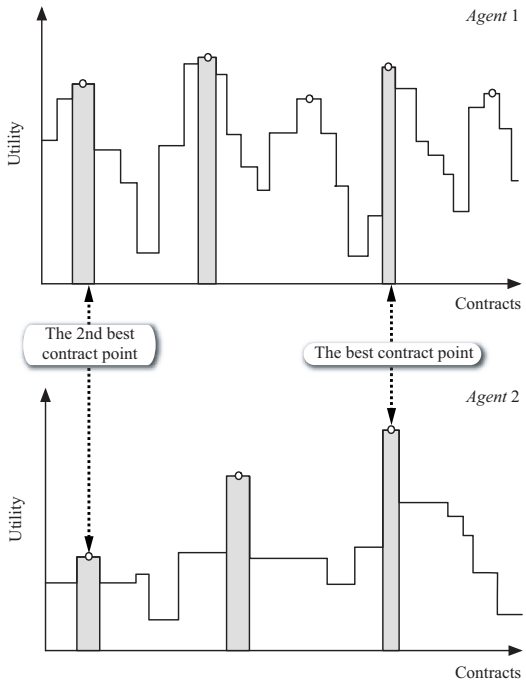


Fig. 4. Deal Identification

Each bidder pays the value of its winning bid to the mediator. The mediator employs breadth-first search with branch cutting to find social-welfare-maximizing overlaps. The detailed algorithm from Step 1 to 4 is shown in the paper [8].

The mediator employs breadth-first search with branch cutting to find social-welfare-maximizing overlaps:
Ag: A set of agents
B: A set of Bid-set of each agent ($B = \{B_0, B_1, ..., B_n\}$, A set of bids from agent i is $B_i = \{b_{i,0}, b_{i,1}, ..., b_{i,m}\}$)

procedure search_solution(B)
$SC := \bigcup_{j \in B_0} \{b_{0,j}\}$, $i := 1$
while $i < |Ag|$ **do**
  $SC' := \emptyset$
  for each $s \in SC$ **do**
    for each $b_{i,j} \in B_i$ **do**
      $s' := s \cup b_{i,j}$
      if s' is consistent **then** $SC' := SC' \cup s'$
  $SC := SC'$, $i := i + 1$
$maxSolution$ = getMaxSolution(SC)
return $maxSolution$

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

bids the agents can generate. Thus, deal identification can terminate in a reasonable amount of time.

In the previous work [8], the threshold for each agent is commonly defined by the mediator. Agents could not change it by their selves. The threshold adjusting mechanism proposed in this paper allows agents to change their threshold values.

3. Incremental Deal Identification

3.1 Threshold Adjusting

In the threshold adjusting mechanism, if an agent reveals the larger area of his utility space, he should gain an advantage. The revealed area can be defined how the agent reveals his utility space on his threshold value. The threshold value is defined at the same value beforehand. Then the threshold values are changed by each agent based on the amount of the revealed area afterwards. If the agent decreases the threshold value, then this means that he reveals his utility space more. This process continues until agents find an agreement. Thus, this negotiation mechanism is incrementally widening the search space.

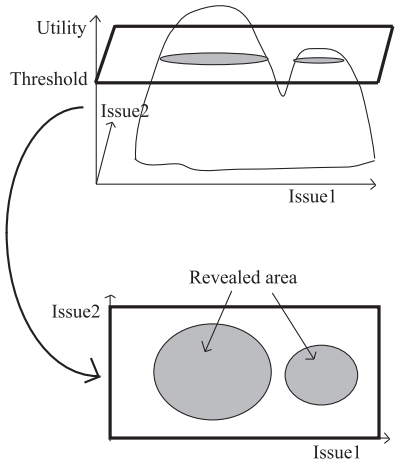


Fig. 5. Revealed Area

Figure 5 shows the concept of "revealed area". If the agent decreases the threshold value, this means he reveals his utility space more. Revealed area when there are more than 2 issues is too complex to give an example of it. It is formulated as follows:

$$r_i = \{\vec{s} \mid u_i(\vec{s}) > Th_i\}$$
$$(r_i : Revealed\ Area,\ Th_i : Threshold\ value)$$

If an agent reveals one point on the grid of utility space, he loses 1 privacy unit in the experiment. If he reveals 1000 points, then he lost 1000 privacy units.

Figure 6 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 the small amount of his utility space. The amount of agent 3's revealing utility space in this threshold adjusting is largest among these 3 agents.

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 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.

The details of the threshold adjusting mechanism is shown as follows:

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

procedure threshold_adjustment( )
loop:
$i := 1$, $B := \emptyset$
while $i < |Ag|$ **do**
bid_generation_with_SA(Th_i, V, SN, T, B_i)
$SC := \emptyset$
$maxSolution$:= search_solution(B)
if $maxSolution$ is not empty
$maxSolution$:= getMaxSolution(SC)
break loop
end if
elseif all agent can lower the threshold
$i := 1$
$SumAr := \Sigma_{i \in |Ag|}\ Ar_i$
while $i < |Ag|$ **do**
$Th_i := Th_i - C * (\sigma_i Ar - Ar_i)/\sigma_i Ar$
$i := i + 1$
end while
end elseif
else
break loop
return $maxSolution$

The above algorithm utilizes step 1, step 2, step 3, and step 4 in the previous section. In the former paper [8], we did not define any external loop of these steps. This paper is the first that proposed the external loop for an effective consenting mechanism.

3.2 Incremental Deal Identification

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

Figure 7 shows the intuitive explanation. The original exhaustive search needs to search the entire space above the bottom (minimum) threshold. On the other hand, our new mechanism does not search the gray area in the figure. Actually, this mechanism might miss the optimal agreement. However, the agreement found by our new mechanism tends to have a higher utility since the threshold initially set on the high utility for each agent. We will check this performance by the experiments.

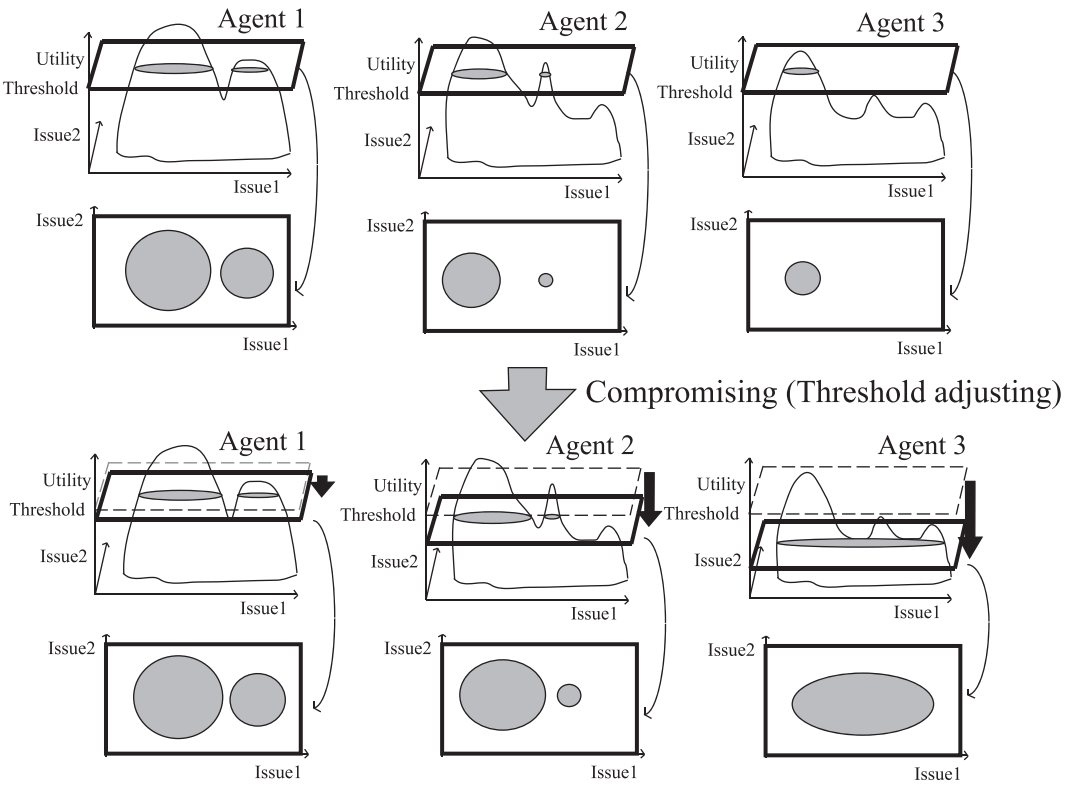


Fig. 6. The Threshold Adjusting Process

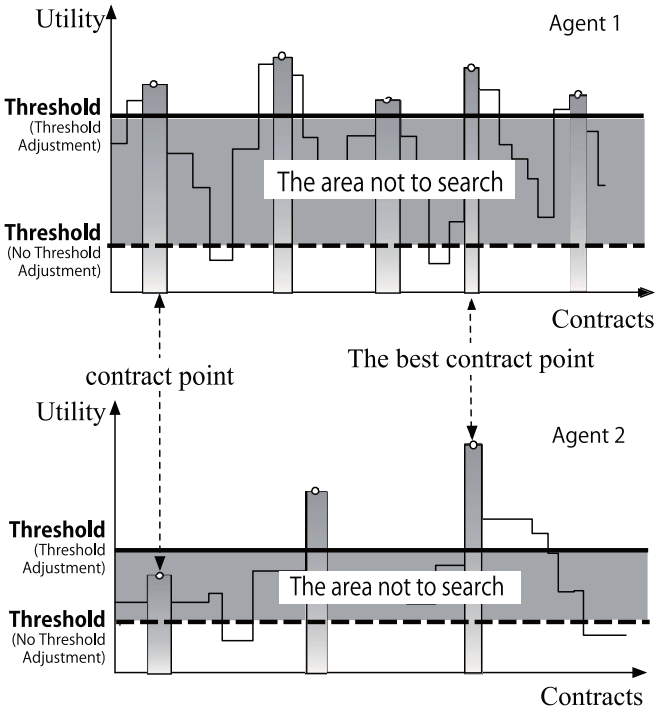


Fig. 7. The reduced search space

4. A Experimental Result

4.1 Setting of Experiments

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

We compare our new threshold adjusting protocol and the existing protocol without adjusting the threshold in terms of optimality and privacy. The following basic setting is same as the paper [8].

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.

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.

The parameters for our experiments were as follows:

- Number of agents is $N = 3$.
- Number of issues is 2 to 10.
- Domain for issue values is $[0, 9]$.
- Constraints include 10 unary constraints, 5 binary constraints, 5 trinary constraints, etc. (a unary constraint relates to one issue, an binary constraint relates to two issues, and so on).
- The maximum value for a constraint is 100 $\times$ (Number of Issues). Constraints that satisfy many issues thus have, on average, larger weights. This seems reasonable for many domains. In meeting scheduling, for example, higher order constraints concern more people than lower order constraints, so they are more important for that reason.

- The maximum width for a constraint is 7. The following constraints, therefore, would all be valid: issue 1 = $[2, 6]$, issue 3 = $[2, 9]$ and issue 7 = $[1, 3]$.
- 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 in starts with 900 and decreases until 200 in the threshold adjusting mechanism.
- The protocol without the threshold adjusting process defines the threshold as 200. The threshold is used to cut out contract points that have low utility.
- The 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: $\sqrt[n]{6400000}$ for n agents. It was only practical to run the deal identification algorithm if it explored no more than about 6400,000 bid combinations, which implies a limit of $\sqrt[n]{6400000}$ bids per agent, for n agents. 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.

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

4.2 Experimental Result

We compared three types of protocols.

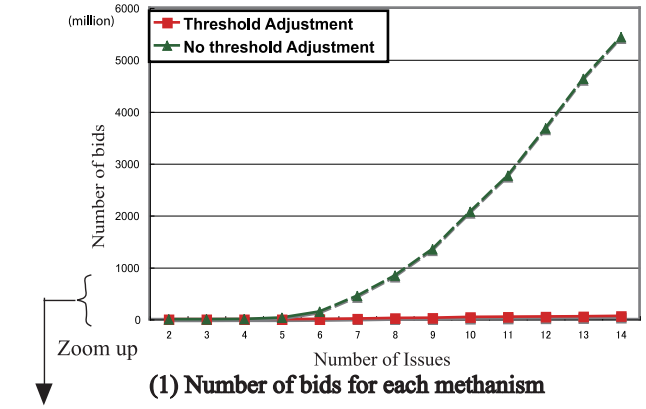
No Threshold Adjustment : The Bidding based multi-issue negotiation protocol is applied [8]. This protocol exhaustively explores the whole utility space.

No Threshold Adjustment with limitation: The Bidding based multi-issue negotiation protocol is applied [8]. This protocol exhaustively explores the whole utility space. However, the number of agent's bids is limited to $\sqrt[n]{6400000}$.

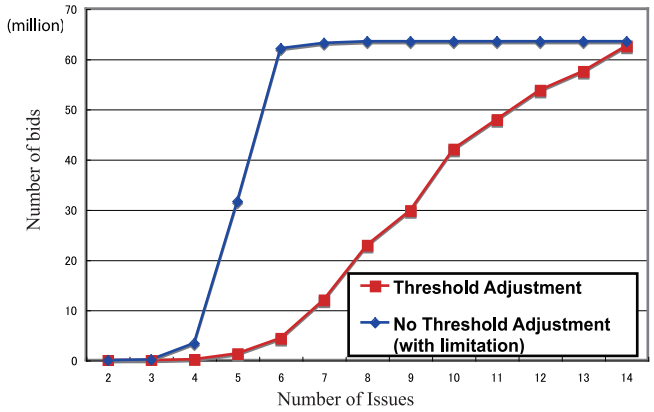
Threshold Adjustment: Our proposed adjusting protocol. This protocol does not have the explicit limitation of the number of bids.

Figure 8 (1) shows the number of bids for each mechanism. The number of bids means the utility space needed to be explored and the time needed to find the possible deal. The number of bids of "No Threshold Adjustment" increases exponentially. Actually, our program fails to compute the combinations completely at more than 6 issues when using "No Threshold Adjustment". Figure 8 (1)just shows the ideal numbers for "No Threshold Adjustment". Our protocol, "Threshold Adjustment", drastically reduces the number of bids.

Figure 8 (2) shows the same results as Figure 8 (1) without "No Threshold Adjustment". "No Threshold Adjustment with limitation" manually limits the number of bids. The increase of the number of bids stops at the limitation defined above. On the other hand, as shown in the Figure 8(2), "Threshold Adjustment" succeeded to reduce the number of bids drastically.



(1) Number of bids for each methanism



(2) Num. of Bids for "Threshold Adjustment" and "No Threshold Adjustment with limitation"

Fig. 8. The number of bids

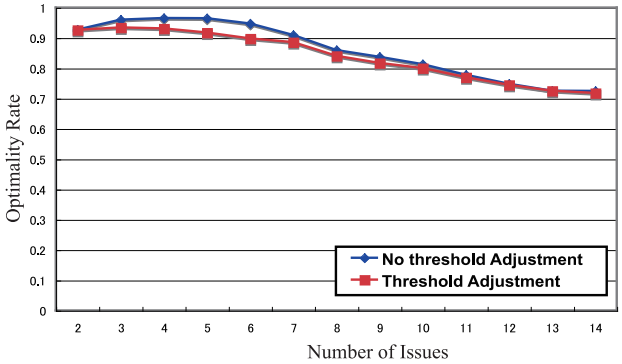


Fig. 9. Optimality

Figure 9 compares the optimality of "No Threshold Adjustment with limitation" and "Threshold Adjustment". Since "Threshold Adjustment" does not search the whole space, its optimality is little less than it of "No Threshold Adjustment with limitation". However, the difference is at most 0.5%. Thus, they are competitive enough.

Fairness on revealed areas is defined as the deviation of the amount of revealed areas for each agent. Thus, to confirm the fairness on revealed areas in our mechanism, we measured average standard deviations on agents' revealed areas. Figure 10 shows the average standard deviations in "No Threshold Adjustment with limitation" and "Threshold Adjustment". Here without loss of generality we assume the number of issues is 3.

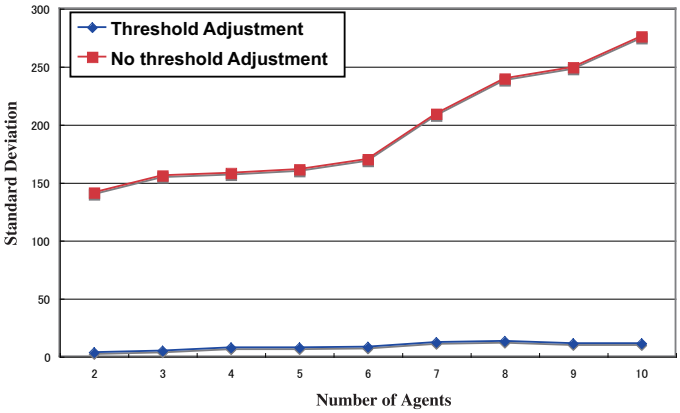


Fig. 10. Standard Deviation

Comparing "Threshold Adjustment" with "No threshold Adjustment" the average standard deviation of the threshold adjustment is much lower than that of the protocol without threshold adjustment. Thus, the threshold adjustment could achieve fair results on the amount of revealed area. Also, the standard deviation increases as the number of agents increases and there are many kinds of agents.

5. Related Work

Some previous works on multi-issue negotiation ([2, 3, 4]) have addressed only linear utilities. A handful of efforts have, however, considered 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 functions is used, nor does it present any experimental analyses. It is therefore unclear whether this strategy enables sufficient exploration of the utility space to find win-win solutions with multi-optima utility functions.

[1] presents an approach based on constraint relaxation. In the proposed approach, a contract is defined as a goal tree, with a set of on/off labels for each goal, which represents the desire that an attribute value is within a given range. There are constraints that describe what patterns of on/off labels are allowable. This approach may face serious scalability limitations. However, there is no experimental analysis and this paper presents only a small toy problem with 27 contracts.

[16] also presents constraint based approach. In this paper, a negotiation problem is modeled as a distributed constraint optimization problem. During exchanging proposals, agents relax their constraints, which express preferences over multiple attributes, over time to reach an agreement. This paper claims the proposed algorithm is optimal, but do not discuss computational complexity and provides only a single small-scale example.

[11] 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.

[12, 13] also presented a protocol for multi-issue problems for bilateral negotiations. [19, 20] presented a multi-item and multi-issue negotiation protocol for bilateral negotiations in electronic commerce situations. [5] proposed a bilateral multi-issue negotiations with time constraints. These studies were done from very interesting view points, but were focusing on just bilateral trading or negotiations.

Our threshold adjusting mechanism can make agreements for two and more agents. Thus, our mechanism is more effective than the methods focusing on just bilateral trading or negotiations ([5],[12],[13][17],[18]). Furthermore, our mechanism demonstrates that the threshold adjusting mechanism can make agreements with very complex utility space in the experiments. Therefore, our mechanism is more realistic and has higher scalability than related works focusing on nonlinear function ([15],[1],[11]). Therefore, our threshold adjusting mechanism is better in some points than other related works.

We focus on the computational aspects in selecting a bid by a mediator. As a reference in the related filed, there are multi-linked negotiation[22]. [22] presents a layered agent framework in which the negotiation process is performed at different levels of abstractions. This approach enables the agent to handle complicated negotiation issues and reduce the computational complexity of searching as the experiment results show. This concept is interesting so, employing the concept of this framework is one of the possible future work to improve the scalability.

As the investigation of the agent's privacy loss, there are Valuations of Possible State (VPS) for distributed constraint optimization (DCOP) ([17],[18]). Using VPS, agents can understand how privacy is lost in the negotiation, scheduling etc. Compared with VPS and our revealed area, revealed area can calculate less computational complexity than VPS. Thus, revealed area is better for the complex negotiation with interdependent issues.

6. Conclusions

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

Possible future work in this area includes improving scalability by developing fast approximate bid generation and deal identification algorithms, as well as by adopting iterative (multi-stage) auction protocols.

References

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

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

[3] P Faratin, C Sierra and N R Jennings, Using similarity criteria

to make issue trade-offs in automated negotiations. Artificial Intelligence 2002, pp. 142:205–237.

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

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

[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. Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS-2007) 2007.

[7] H Hattori, M Klein and T Ito, A multi-phase protocol for negotiation with interdependent issues. Proceedings of the International Conference on Intelligent Agent Technology (IAT/WI 2007) 2007.

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

[9] T Ito and M Klein, A multi-issue negotiation protocol among competitive agents based on auctions. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2006) 2006.

[10] T Ito, M Klein and H Hattori, A negotiation protocol for agents with nonlinear utility functions. Proceedings of 21st National Conference on Artificial Intelligence (AAAI2006) 2006.

[11] M Klein, P Faratin, H Sayama and Y Bar-Yam, Negotiating complex contracts. Group Decision and Negotiation, Vol. 12, No. 2, 2003, pp. 58–73.

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

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

[14] R R K Lau, Towards genetically optimised multi-agent multi-issue negotiations. Proceedings of HICSS-2005 2005.

[15] R J Lin and S T Chou, Bilateral multi-issue negotiations in a dynamic environment. Proceedings of AMEC-2003 2003.

[16] 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, Vol. 148, 2003, pp. 53–102.

[17] R T Maheswaran, J P Pearce, P Varakantham, E Bowring and M Tambe, Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. Autonomous Agents and Multi-Agent Systems, Vol. 13, 2008, pp. 27–60.

[18] J P Rachel Greenstadt and M Tambe, Analysis of privacy loss in distributed constraint optimization. Proceedings of the 17th Innovative Applications of Artificial Intelligence Conference 2006.

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

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

[21] L K Soh and X Li, Adaptive, confidence-based multiagent negotiation strategy. Proceedings of Third International Joint Conference on Autonomous Agent and Multi-Agent Systems (AAMAS-2004) 2004, pp. 1048–1055.

[22] X Zhang, V Lesser and T Wagner, A layered approach to complex negotiations. Web Intelligence and Agent System, Vol. 2, No. 2, 2004, pp. 91–104.