Automated Mediation Protocols based on Monotonic Tree Representations

Katsuhide Fujita[1,a)]

Abstract: Automated negotiations occur when a negotiating function is performed among intelligent agents. Although current human-to-human negotiation appears to involve multiple extremely complex issues, each automated negotiation setting is simple. In particular, the structure of issues is independent and flat in the existing automated negotiation framework. In this paper, we propose realistic negotiation frameworks for non-monotonic utility functions. The monotonicity of the utility functions is an important characteristic because if the utility function is monotonic, the issues are independent. When the issues are independent, it is useful to separate them and reach a distinct agreement for each sequentially. In addition, we propose an automated mediation protocol for multiple non-monotonic issue negotiations. This mediation protocol consists of the communications between agents and the mediator. The procedures of the mediation protocol include recognizing related issues, announcement, bidding, awarding, and expediting. We experimentally demonstrate that the proposed method results in good outcomes and greater scalability. In addition, we demonstrate that a suitable mediation strategy leads to better outcomes and scalability.

Keywords: automated multi-issue negotiation, agreement technology, monotonic utility function

1. Introduction

Automated negotiation is an important aspect of daily life and represents an important topic in the field of multiagent system research. There has been extensive work in the area of automated negotiation, in which automated agents negotiate with other agents in contexts such as e-commerce [1] and large-scale deliberation [2]. Automated negotiations occur when the negotiating function is performed among intelligent agents. Although current human-to-human negotiation appears to involve multiple extremely complex issues, most of the existing work on automated negotiation settings is simple (Refs. [3], [4] etc.). For example, the structure of issues is independent and flat.

A key point in achieving automated negotiation frameworks in complex situations is the non-monotonicity of the utility functions. If the utility function is monotonic, the issues are independent and not interdependent. Many real-world negotiation problems involve multiple interdependent issues. When designers work together to design a car, for example, the value of a given carburetor depends strongly on which engine is chosen. When an automated negotiation covers multiple independent issues, it is useful to separate them and reach a separate agreement on each sequentially. However, this is not always possible or desirable because one issue affects another.

Recently, some papers attempted to consider the interdependence of issues and non-monotonicity. For example, Robu et al. [6] propose the utility graph, which captures allocation preferences as a set of nodes (each representing whether a given good

was purchased). In addition, Ito et al. [7], Lopez-Carmona et al. [8], and Fujita et al. [9] focus on constraints-based utility functions, which are highly nonlinear and bumpy. In this paper, we propose a novel representation for non-monotonic utility functions, which leads to an efficient negotiation protocol.

First, we propose a novel monotonic tree structure to detect the non-monotonic relationships between issues. The monotonic tree is based on the term trees, and its branches represent the monotonic relation. The leaves of the tree represent the non-monotonic term set. By using the monotonic tree, we can distinguish effective issue grouping in which the complexity of finding contracts is low and social welfare is high. Rosenschein et al. [10] and Chevaleyre et al. [11] have also explored the idea of tree-structured domains in the context of combinatorial auctions or resource allocation. However, these papers don't employ tree-structured domains to represent monotonicity.

Next, we propose a novel automated negotiation protocol in which the mediator tries to reorganize a highly complex utility space into several tractable utility subspaces, in order to reduce computational cost. This mediation protocol consists of communications between agents and the mediator. The mediation protocol procedures include recognizing related issues, announcement, bidding, awarding, and expediting. Using a mediator allows us to employ the knowledge sharing protocol (e.g., sharing the monotonic tree) and effectively find points of agreement.

In addition, the adjustment step is a key point for effective automated negotiations. This is because of the trade-off between social welfare and the complexity of the consensus. If the level of the monotonic tree is too high, the consensus between agents is too complex. Agents usually have constraints regarding issue relationships; however, these constraints sometimes disturb

---

1   Faculty of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184–8588, Japan
a)   katfuji@cc.tuat.ac.jp

the consensuses [4], [12]. On the other hand, if the level of the monotonic tree is too low, the mediator misses important interdependencies between the issues. Going down the monotonic tree means that the mediator ignores some interdependency, reducing social welfare. In this paper, we propose two methods for adjusting the height level: the tree-climbing protocol and the tree-descending protocol.

Finally, we demonstrate that our protocol has a higher optimality rate and discuss the impact of the negotiation outcomes on optimality. In addition, the protocol is influenced by the height of the monotonic tree. We also analyze our protocol on the basis of the experimental results.

The remainder of this paper is organized as follows: We first describe a multi-issue negotiation model, monotonic utility functions, and the monotonic tree. Next, we present the basic mediation protocol on the basis of monotonic trees, as well as techniques for adjusting the monotonic tree height. We then present the experimental results, demonstrating that our protocol produces more optimal outcomes. Finally, we describe related work and present our overall conclusions.

2. Automated Negotiations and Monotonic Utility Functions

2.1 Multi-issue Negotiation Environments

Negotiation in multiagent systems is the process by which a group of agents come to a mutually acceptable agreement on some matter [4]. Negotiation underpinnings attempt to cooperate and coordinate when the agents are self interested and when they are cooperative [13]. For modeling a negotiation situation with multiple hierarchal issues, we define the players, negotiation issues, and objects in the negotiation.

Definition 1 Agents and mediator: N agents $(a_1, \ldots, a_N)$ want to reach an agreement with a mediator who manages the negotiation from a man-in-the-middle position. Agents act in line with the negotiation protocol to achieve their objectives. Mediators also act in line with the protocol to support consensus building. In this paper, the mediator is not self-oriented or trying to achieve a specific benefit.

Definition 2 Issues under negotiation: There are M issues $(i_1, \ldots, i_M)$ to be negotiated.

Definition 3 Contract space: The negotiation solution space is defined by the values that the different issues may take. To simplify, we assume that the issue takes a value drawn from the domain of integers $\{0, 1, ..., X_i\}$

Definition 4 Alternative to contract or potential solution: $\vec{s} = (s_1, ..., s_M)$. A contract is represented by a vector of issue values.

Definition 5 Utility function for agent a: $u_a(\vec{s})$.

Definition 6 Objective functions: $\arg\max_{\vec{s}} \sum_{a \in N} u_a(\vec{s})$. $u_a(\vec{s}) > \delta$, $(a = 1, \ldots, N)$ (δ is the reservation value). Our protocol, in other words, tries to find contracts that maximize social welfare, i.e., the total utility for all agents. Such contracts, by definition, will also be Pareto optimal. Simultaneously, each agent tries to find contracts where its individual welfare is more than the reservation value. The reservation value is the lowest value the agent can accept for the contracts.

2.2 Monotonic Utility Function

The monotonicity of the utility functions is an important characteristic because it guarantees that the issues are independent. The monotone theory appears in some articles on the subject, which give examples from special applications. In addition, utility theory also appears in some articles such as those on combinational auctions [14] or allocation theory [15]. In this subsection, we define the monotonic utility function and describe its important features.

Definition 7 Monotonicity of utility function for a multivariable function: A utility function $u_a(s_1, s_2, \ldots, s_k)$ is said to be a *monotonic* if and only if s_k and s'_k, $s_k \geq s'_k$ implies $u_a(s_1, s_2, \ldots, s_k) \geq u_a(s'_1, s'_2, \ldots, s'_k)$ in $\forall k$ for agent a.

In other words, $u_a(s_1, s_2, \ldots, s_k)$ is maximum *if* utility function u_a is *monotonic* and $s_1, s_2, \ldots, s_k$ are maximum for agent a.

Using this definition, we can obtain sub-negotiation protocols according to the issues regardless of whether the issues are independent. Many real-world negotiation problems involve multiple interdependent issues. When designers work together to design a car, for example, the value of a given carburetor depends strongly on which engine is chosen. In multiple *independent* issues negotiation, it is easy to divide the negotiation issues and find the optimal point for each issue. For example, say there are three issues in the negotiation, and the mediator wants to find the optimal contract. In multiple *independent* issue negotiation, the mediator finds the optimal contract for each issue and adds the maximum utility values for each issue. On the other hand, the mediators cannot find the optimal contract for each issue easily in multiple *interdependent* issue negotiation, because the utility value of an individual issue is influenced by other issues. Thus, the mediator needs to search exhaustively despite the large computational cost.

2.3 Monotonic Tree

We define the Issue Interdependent Tree for representing the monotonicity of the utility functions. The set of terms of a utility function u_a is the set of bundles T with a nonzero coefficient α^T. For instance, the utility function $u_a(\vec{s}) = 16.s_2 + 5.s_1.s_3.s_4 + 2.s_1.s_3$ uses the terms s_2, $s_1.s_3.s_4$, and $s_1.s_3$. If the functions are monotonic, + is used, and if they are non-monotonic, . is used. In the example, i_1, i_3, and i_4 are non-monotonic, and other relationships are monotonic for agent a. T will denote the set of all terms appearing in the representation of any of the utility functions $(u_1, \ldots, u_N)$, and α^T will denote the coefficient of term T in u_a. Finally, T^l denotes the set of terms in T consisting of exactly l issues, and $T^{\leq l}$ denotes the set of terms in T with at most l non-monotonic functions each.

Intuitively, tree-structured utilities are k-additive functions in which there are no "overlapping" terms.

Definition 8 A set of utility functions $\{u_1, \ldots, u_N\}$ is called tree-structured *if*, if it is a case in which all terms $T_1, T_2 \in T$ have either $T_1 \subseteq T_2$ or $T_1 \supseteq T_2$ or $T_1 \cap T_2 = \{\}$.

In the monotonic tree, the T terms can be represented by a tree in which R is the root, and each term is a node. The branches of the tree represent the "monotonic" relation, and the leaves of the tree represent the non-monotonic terms set. The following example illustrates this representation:
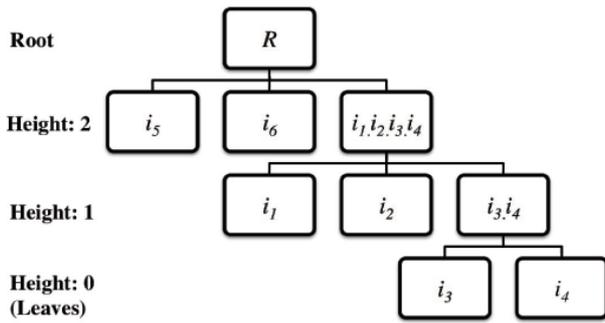
Fig. 1   Example of monotonic tree.

Example 1   Agent 1, Agent 2, and Agent 3 have utility functions u_1, u_2, and u_3, as follows:

$$u_1(\vec{s}) = s_2 + 3s_5$$

$$u_2(\vec{s}) = 3s_1 + 10s_1.s_2.s_3.s_4 + 8s_5 + 4s_6$$

$$u_3(\vec{s}) = s_6 + s_4 + 8s_3.s_4$$

In this situation, there are six issues $(i_1, \ldots, i_6)$. The set of issues $T = \{i_1, i_2, i_4, i_5, i_6, i_3.i_4, i_1.i_2.i_3.i_4\}$ can also be represented by **Fig. 1**.

The height of the tree (l) represents how many agents agree with the monotonicity between the issues. For example, Agent 3 agrees with the monotonicity when "$l = 1$." This is because issue 3 and issue 4 are non-monotonic in the utility function of Agent 3. On the other hand, Agent 2 does not agree with the monotonicity because issue 1, issue 2, issue 3, and issue 4 are non-monotonic in its utility function.

Rosenschein et al. [10] and Chevaleyre et al. [11] have also explored the idea of tree-structured domains in the context of combinatorial auctions or resource allocations. However, these papers don't address monotonicity.

3.   Automated Mediation Protocols Based on Monotonic Tree

3.1   Baseline Mediation Protocol

In this paper, we propose a novel approach in which agents reach an agreement. The proposed protocol is a remarkable result focusing on automated negotiation with non-monotonicity. In this protocol, many agents (participants) and a mediator appear. **Figure 2** shows the flow of the automated mediation protocol. The proposed automated negotiation protocol consists of the following steps:

Step 1:   Recognizing the Grouping Issues: In this step, the mediator identifies the effective issue groups. First, the mediator generates the monotonic tree. Next, it identifies the issue groups on the basis of the tree. Identifying the issue groups is not very difficult. The mediator gathers all nodes that have a height of "l." Each node's terms indicate effective issue grouping.

Step 2:   Announcement: The mediator sends out an announcement to the agents about submitting bids. The announcement includes the issue grouping information described in the previous step.

Step 3:   Bidding: Each agent generates a bid by searching the utility functions. For each contract $\vec{s}$ found through an exhaustive
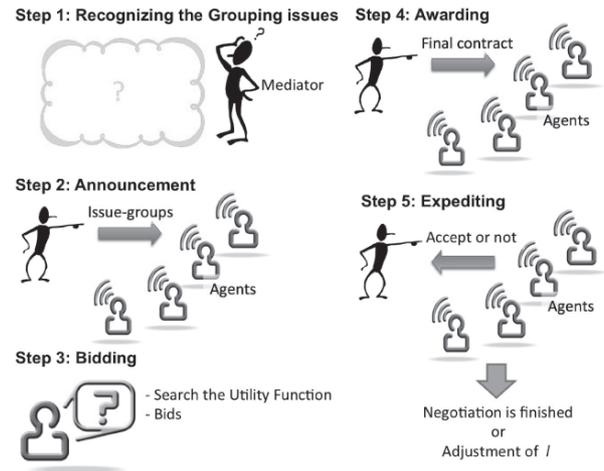

Fig. 2   Flow of baseline mediation protocol.

search, an agent evaluates the utility on the basis of the utility function. If that utility is larger than the reservation value δ, the agent defines a bid that has a domain set and the utility value. The reservation value is the lowest value agents can accept for the contracts. Next, the agents divide the bids into those for each issue group and set the evaluation values for these bids.

Step 4:   Awarding: After the mediator sends the contract announcement, it must choose among the received bids and decide which contract is awarded. The mediator identifies the final contract by finding the bid combinations, one from each agent, that are mutually consistent. 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). The result of this process is communicated to the agents that submitted a bid.

Step 5:   Expediting: After the agents receive the contract, they indicate whether it is accepted. If all agents accept the alternative, the negotiation is finished. If at least one agent does not accept the alternative, the mediator adjusts l based on the adjustment method, which is described in the next subsection.

The basic idea of this protocol has its roots in the Contract Net Protocol [16]. The roles of the manager and agents are similar to those of the mediator and agents in our protocol. By employing a mediator, we can use the shared protocol knowledge (e.g., sharing the monotonic tree) and effectively identify the points of agreement.

3.2   Adjusting Height Level of Monotonic Tree

For an effective automated negotiation protocol in non-monotonic situations, the adjustment step is a key point. This is because there is a trade-off between social welfare and the complexity of the consensus. If the height level of the monotonic tree is too high, the consensus between agents is too complex. Agents usually have issue relationship constraints; however, these constraints sometimes disturb the consensuses. On the other hand, if the height level of the monotonic tree is too low, the mediator misses the important interdependency between the issues. Going down the monotonic tree means that the mediator ignores some

interdependency, thus reducing social welfare.

In this paper, we propose two methods for adjusting the height level: the tree-climbing protocol and the tree-descending protocol. The basic idea of tree-climbing protocols is to allow T^0 involving only the smallest groups (the size of groups is only 1) first, and then incrementally allow bigger bundles until the mediator reaches an agreement. In addition, the automated negotiation protocol in the previous section is carried out, modifying the utility functions of each agent during each round of the protocol. Algorithm 1 shows the tree-climbing protocol.

Algorithm 1 Tree-climbing protocol

---

n agents with monotonic tree. ($u_1, \ldots, u_n$: Utility functions; T^l: the set of all terms in the tree when the height level of the tree is l, $|T|$: the height of the tree.)

1: $l := 0$
2: **while** $l < |T|$ **do**
3:     Restrict allowed deals to T^l.
4:     Let mediator and agents negotiate T^l.
5:     **if** agents cannot make agreements **then**
6:         $l := l + 1$
7:     **end if**
8: **end while**

The basic idea of tree-descending protocols is to allow T^M involving only the largest group including all issues first, and then to incrementally allow smaller bundles until the mediator reaches an agreement. In addition, the automated negotiation protocol in the previous section is carried out, modifying the utility functions of each agent during each round of the protocol. Algorithm 2 shows the tree-descending protocol.

4. Experimental Results

4.1 Setting

We conducted several experiments to evaluate our approach. In each experiment, we ran 100 negotiations. The following parameters were used: The domain for the issue values is $\{0, 1, \ldots, 9\}$. The utility of each domain is chosen randomly from $\{0, 1, \ldots, 10\}$. We use following utility function for expressing the non-monotonic functions. If $u(a, b)$ is a monotonic function, $u(a, b) = (u(a) + u(b))/2$. If $u(a, b)$ is a non-monotonic function, $u(a, b) = |10 \sin(a + b)|$. By using this utility functions, the utility values are normalized from 1 to 10.

We compared the following negotiation methods:

Algorithm 2 Tree-descending protocol

---

n agents with monotonic tree. ($u_1, \ldots, u_n$: Utility functions; T^l: the set of all terms in the tree when the height level of the tree is l, $|T|$: the height of the tree.)

1: $l := |T|$
2: **while** $l > 0$ **do**
3:     Restrict allowed deals to T^l.
4:     Let mediator and agents negotiate T^l.
5:     **if** agents cannot make agreements **then**
6:         $l := l - 1$
7:     **end if**
8: **end while**

- "(A) Tree-climbing protocol" applies the tree-climbing protocol to the automated negotiation protocol proposed in this paper.
- "(B) Tree-descending protocol" applies the tree-descending protocol to the automated negotiation protocol proposed in this paper.
- "(C) One-shot Automated Negotiation Protocol (Random)" applies the automated negotiation protocol proposed in this paper without monotonic tree height adjustment. The height l is chosen randomly.
- "(D) One-shot Automated Negotiation Protocol (Root)" applies the automated negotiation protocol proposed in this paper without monotonic tree height adjustment. The height l is the maximum value (Root).
- "(E) One-shot Automated Negotiation Protocol (Leaves)" applies the automated negotiation protocol proposed in this paper without monotonic tree height adjustment. The height l is 0 (Leaves).

We applied a centralized exhaustive search for the sum of the individual agents' utility functions to obtain the optimal social welfare for each negotiation test run. We calculated a normalized optimality rate for each negotiation run, defined as *(social welfare achieved by each protocol)/(optimal social welfare calculated by Simulated Annealing)*. This central Simulated Annealing method is generally unrealistic in negotiation contexts because it requires that agents fully reveal their utility functions to a third party. The failure rate for each negotiation run is defined as *(the number of successful negotiations)/100*. If the number of issues in the experiments varies, the number of agents is 5 and the reservation value is 0.2. If the number of agents varies, the number of issues is 20 and the reservation value is 0.2. If the reservation value varies, the number of issues is 20 and the number of agents is 5.

Our code was implemented in Java 2 (1.6) and was run on a core-i7 with 4.0 GB of memory under Mac OS X (10.6).

4.2 Results

Figure 3 compares the optimality rate when the number of issues or agents varies. "(D) One-shot automated negotiation protocol (Root)" achieved the highest optimality rate. This is because it accepts only the optimal solutions; therefore, it is not robust with respect to scalability of negotiation problems. The results of "(A) Tree-climbing protocol" are almost the same as those of "(E) One-shot automated negotiation protocol (Leaves)." This means that (A) does not change the height of the monotonic tree. In other words, it finds agreements at the initial step even though some of the interdependency between issues is ignored. "(B) Tree-descending protocol" yields the second-highest optimality rate, and higher than the "(C) One-shot automated negotiation protocol (Random)." This means that (B) finds the contracts and obtains an effective height for the monotonic tree.

Figure 4 compares the failure rate when the number of issues or agents varies. All the protocols except for "(C) One-shot automated negotiation protocol (Random)" and "(D) One-shot automated negotiation protocol (Root)" can reach agreement. This means that the proposed adjustment protocol can work well to find agreements. "(E) One-shot automated negotiation protocol
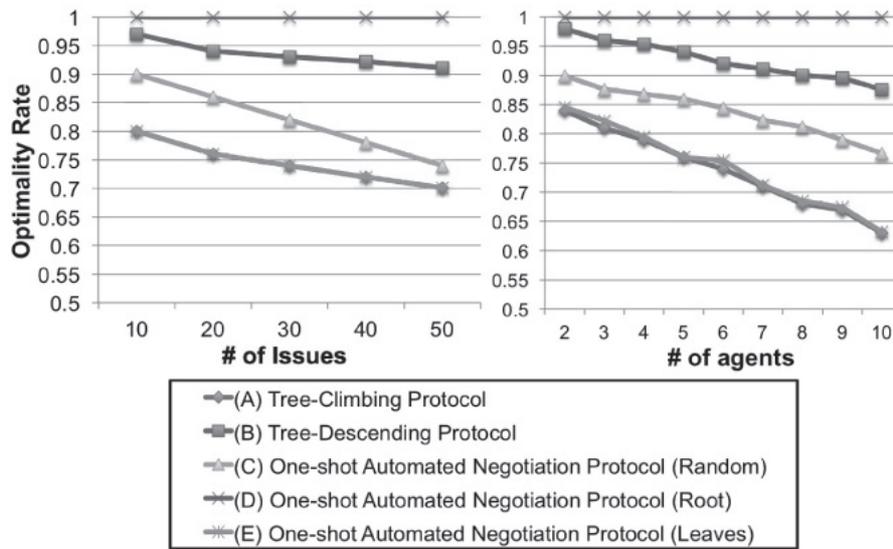
Fig. 3 Comparison of optimality when the number of issues or agents varies.
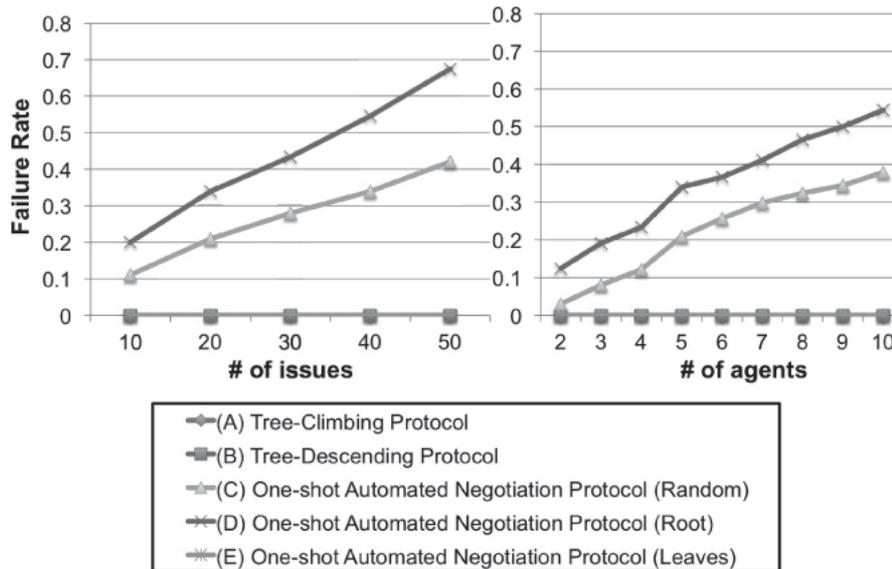

Fig. 4 Comparison of failure rate when the number of issues or agents varies.

(Leaves)" reaches agreement without adjusting the height of the monotonic tree because it is easy to identify agreements at the bottom of the tree.

Figure 5 compares the optimality and failure rate for various reservation values. The reservation value indicates the selfishness of the agents. If the reservation value is high, it is hard to make agreements because of the selfishness of the agents. When the reservation value increases, the optimality rate decreases. This is because the agents don't concur with the solutions having high social welfare. In addition, the failure rate is high when the reservation value becomes high. This is because agents don't agree with solutions that usually produce agreements between unselfish agents.

5.　Related Work

Klein et al. [17] presented a protocol that produces near-optimal results in medium-sized bilateral negotiations with binary dependencies, but it was not applied to multilateral negotiations and higher-order dependencies. In addition, Ito et al. [7] presented a bidding-based protocol aimed at complex utility spaces, where agents generate bids by finding high regions in their own utility functions, and the mediator determines the optimum combination of bids submitted by the agents. Lopez-Carmona et al. [8] proposed a novel auction-based protocol using weighted constraints and addressing highly rugged utility spaces. These existing works tackle similar issues in automated negotiation frameworks. However, they don't use the concept of the monotonic tree proposed in this paper. Fujita et al. [18], [19] propose the effective negotiation protocols based on the nonhierarchical grouping.

Jonker et al. [5] propose a model for bargaining with incomplete information, while Robu et al. [6], [20] propose a utility graphs formalism to address complex multi-issue negotiations.
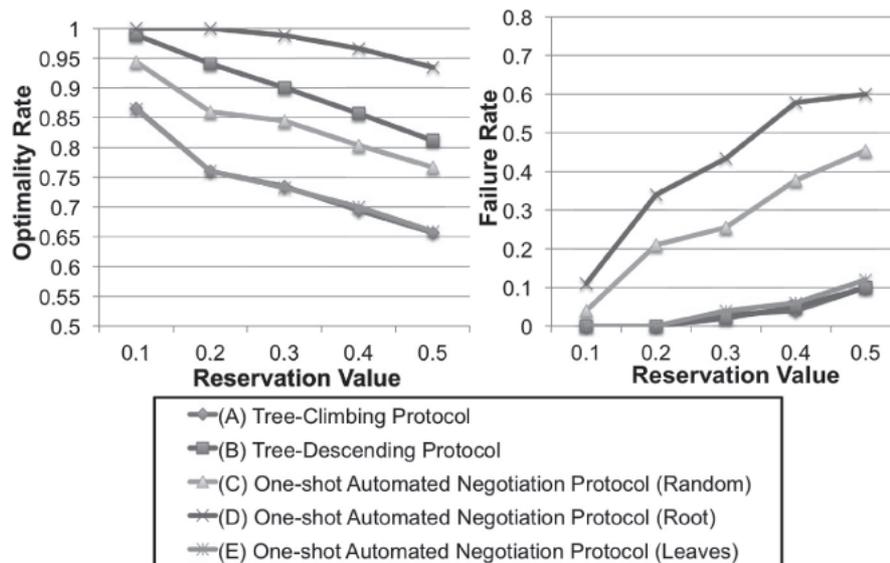
Fig. 5 Comparison of optimality and failure rate for various reservation values.

The utility graph captures allocation preferences as a set of nodes (each representing the issue of whether a given good was purchased), in addition to a set of links between these nodes that capture the (positive or negative) complementarities between the goods. The utility graph also shows the interdependency (monotonicity) between issues using the graph. In this paper, we use a tree structure containing multiple hierarchal issues, not a graph structure.

CP-nets capture preferential dependencies for N-ary issues using directed graphs in which each node represents the agent's preference for an issue, and each link captures the impact of one issue choice on the preferences for another [21]. A CP-net focuses on a network structure, not a tree structure. By introducing the tree structure, we could propose a method for searching tree structures in complex automated negotiations.

In addition, some promising approaches have been suggested in recent years. Hindriks et al. [22] proposed an approach based on a weighted approximation technique to simplify the utility space. The resulting approximated utility function can be handled by negotiation algorithms designed for multiple independent issues and has a polynomial time complexity. Our protocol can find an optimal agreement point if agents don't have an expected negotiation outcome in common. An et al. [23] proposed the design and implementation of a negotiation mechanism for dynamic resource allocation problems in cloud computing. Multiple buyers and sellers are allowed to negotiate with each other concurrently, and an agent is allowed to decommit from an agreement only after paying a penalty.

Lin et al. [24], [25] focused on Expert Designed Negotiators, which are based on negotiations between humans and automated agents in real life. In addition, tools for evaluating automatic agents that negotiate with people are proposed. These studies include some demonstrations of efficient negotiations in extensive experiments involving many human subjects and personal digital assistants. However, these studies don't consider the non-monotonic utility function, on which this paper focuses.

Fenghui et al. [26] proposes a bilateral single-issue negotiation model for nonlinear utility functions. A multiple offers mechanism is introduced to handle non-monotonic utility functions, and an approximating offer mechanism is introduced to handle discrete utility functions. Our paper focuses on multiple issues and agents negotiations, not only a bilateral single-issue negotiations.

6. Conclusions

In this paper, we focused on the non-monotonicity of utility functions and proposed the monotonic tree for recognizing effective issue grouping. The monotonicity of the utility functions is an important characteristic because it guarantees that the issues are independent. In addition, we proposed an automated mediation protocol for multiple non-monotonic issue negotiation. This mediation protocol consists of the communications between agents and a mediator. The procedures in the mediation protocol include recognizing related issues, announcement, bidding, awarding, and expediting. We demonstrated experimentally that the proposed method permits good outcomes and greater scalability. In addition, we demonstrated that a suitable mediation strategy leads to better outcomes and scalability.

In our future work, we will address incentive compatibility issues. In the bilateral case, we found that this can be done using a type of Clarke tax [27], wherein each agent has a limited budget from which it has to pay other agents before the mediator accepts a contract that favors that agent but reduces the utility for the others. This approach gives agents the incentive to avoid exaggeration, because exaggerating will cause them to spend their limited budget on contracts that don't strongly affect their true utility values. We will investigate whether and how this approach can be applied to the multilateral case.

References

[1] Kraus, S.: *Strategic Negotiation in Multiagent Environments*, Cambridge University Press (2001).

[2] Malone, T.W. and Klein, M.: Harnessing Collective Intelligence to Address Global Climate Change, *Innovations Journal*, Vol.2, No.3, pp.15–26 (2007).

[3] Faratin, P., Sierra, C. and Jennings, N.R.: Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations, *Artificial Intelligence*, pp.142:205–237 (2002).

[4] Luo, X., Jennings, N.R., Shadbolt, N., Leung, H. and Lee, J.H.: A Fuzzy Constraint based Model for Bilateral, Multi-issue Negotiations in Semi-competitive Environments, *Artificial Intelligence*, Vol.148, pp.53–102 (2003).

[5] Jonker, C.M., Robu, V. and Treur, J.: An Agent Architecture for Multi-Attribute Negotiation Using Incomplete Preference Information, *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, Vol.15, pp.221–252 (2007).

[6] Robu, V. and Poutre, H.L.: Retrieving the Structure of Utility Graphs Used in Multi-Item Negotiation through Collaborative Filtering of Aggregate Buyer Preferences, *Proc. 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)* (2006).

[7] Ito, T., Hattori, H. and Klein, M.: Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces, *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pp.1347–1352 (2007).

[8] Lopez-Carmona, M., Marsa-Maestre, I., Klein, M. and Ito, T.: Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces, *Autonomous Agents and Multi-Agent Systems*, Vol.24, No.3, pp.485–535 (2012).

[9] Fujita, K., Ito, T. and Klein, M.: A Secure and Fair Protocol that Addresses Weaknesses of the Nash Bargaining Solution in Nonlinear Negotiation, *Group Decision and Negotiation*, Vol.21, pp.29–47 (2012).

[10] Rosenschein, J.S. and Zlotkin, G.: *Rules of Encounter*, MIT Press (1994).

[11] Chevaleyre, Y., Endriss, U. and Maudet, N.: Tractable negotiation in tree-structured domains, *Proc. 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2006)*, pp.362–369 (2006).

[12] Luo, X., man Lee, J.H., fung Leung, H. and Jennings, N.R.: Prioritised Fuzzy Constraint Satisfaction Problems: Axioms, Instantiation and Validation, *Fuzzy Sets and Systems*, Vol.136, pp.151–188 (2003).

[13] Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C. and Wooldridge, M.: Automated Negotiation: Prospects, Methods and Challenges, *International Journal of Group Decision and Negotiation*, Vol.10, No.2, pp.199–215 (2001).

[14] Lai, J. and Parkes, D.: Monotone branch-and-bound search for restricted combinatorial auctions, *Proc. 13th ACM Conference on Electronic Commerce (EC '12)*, pp.705–722 (2012).

[15] Landsberger, M. and Meilijson, I.: Co-monotone allocations, Bickel-Lehmann dispersion and the Arrow-Pratt measure of risk aversion, *Annals of Operations Research*, Vol.52, pp.97–106 (1994).

[16] Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. Comput.*, Vol.29, No.12, pp.1104–1113 (1980).

[17] Klein, M., Faratin, P., Sayama, H. and Bar-Yam, Y.: Negotiating Complex Contracts, *Group Decision and Negotiation*, Vol.12, No.2, pp.58–73 (2003).

[18] Fujita, K., Ito, T. and Klein, M.: An Approach to Scalable Multi-issue Negotiation: Decomposing the Contract Space, *Computational Intelligence* (online), DOI: 10.1111/j.1467-8640.2012.00462.x (2012).

[19] Fujita, K., Ito, T. and Klein, M.: Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents, *Decision Support Systems* (online), DOI: 10.1016/j.dss.2013.05.016 (2013).

[20] Robu, V., Somefun, D.J.A. and Poutre, J.L.: Modeling complex multi-issue negotiations using utility graphs, *Proc. 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pp.280–287 (2005).

[21] Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H. and Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements, *Artificial Intelligence Research*, Vol.21, pp.135–191 (2004).

[22] Hindriks, K., Jonker, C. and Tykhonov, D.: Eliminating Interdependencies Between Issues for Multi-issue Negotiation, *Cooperative Information Agents X, Lecture Notes in Computer Science*, Vol.4149, pp.301–316 (2006).

[23] An, B., Lesser, V.R., Irwin, D. and Zink, M.: Automated negotiation with decommitment for dynamic resource allocation in cloud computing, *Proc. 9th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pp.981–988 (2010).

[24] Lin, R. and Kraus, S.: Can automated agents proficiently negotiate with humans?, *Comm. ACM*, Vol.53, No.1, pp.78–88 (2010).

[25] Lin, R., Kraus, S., Oshrat, Y. and Gal, Y.K.: Facilitating the Evaluation of Automated Negotiators using Peer Designed Agents, *Proc. 24th Association for the Advancement of Artificial Intelligence (AAAI-2010)* (2010).

[26] Ren, F. and Zhang, M.: Bilateral Single-Issue Negotiation Model Considering Nonlinear Utility and Time Constraint, *Decision Support Systems* (online), DOI: 10.1016/j.dss.2013.05.018 (2013).

[27] Sandholm, T.W.: *Distributed Rational Decision Making*, Multi-Agent Systems, Weiss, G. (Ed.) (1998).

Katsuhide Fujita received his B.E., M.E, and Ph.D. degrees from Nagoya Institute of Technology in 2008, 2010, and 2011, respectively. From 2010 to 2011, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). From 2010 to 2011, he was a visiting researcher at MIT Sloan School of Management. From 2011 to 2012, he was a Project Researcher of School of Engineering, the University of Tokyo. He is an Associate Professor of Faculty of Engineering, Tokyo University of Agriculture and Technology since 2012. His main research interests include multi-issue negotiation, multi-agent systems, intelligent agents and decision support systems. His research interest is online publishing systems. He is a member of the IEEE, AAAI, IPSJ, JSAI, and IEICE.