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, 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
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 hierarchical issues, we define the players, negotiation issues, and objects in the negotiation.

Definition 1 Agents and mediator: N agents (a1,...,aN) 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 (i1,...,iM) 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,...,Xi}

Definition 4 Alternative to contract or potential solution: s̃ = (s̃1,..., s̃M). A contract is represented by a vector of issue values.

Definition 5 Utility function for agent a: ua(s̃)

Definition 6 Objective functions: arg max a \sum N \in N ua(s̃), ua(s̃) > δ, (a = 1,...,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 combinatorial 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 multi-variable function: A utility function ua(s1,s2,...,sk) is said to be a monotonic function if and only if sα and s′α, sβ ≥ s′β implies ua(s1, s2, ..., sk, sα) ̸= ua(s1, s2, ..., sk, s′α) in ∀ k for agent a.

In other words, ua(s1,s2,...,sk) is maximum if utility function ua is monotonic and s1, s2,...,sk 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 ua is the set of bundles T with a nonzero coefficient aT. For instance, the utility function ua(δ) = 16.s2+5.s1.s3+2.s1.s3 uses the terms s2, s1.s3, s4, and s1.s3. If the functions are monotonic, + is used, and if they are non-monotonic, . is used. In the example, i1, i3, and i4 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 (ua1,...,uAN), and aT will denote the coefficient of term T in ua. Finally, Tl denotes the set of terms in T consisting of exactly l issues, and Td 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 {ua1,...,uAN} is called tree-structured if, if it is a case in which all terms Ti1,Ti2 ∈ T have either T11 ⊆ T12 or T11 ⊇ T12 or T11 ∩ T12 = {}.

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:
In this situation, there are six issues \( (i_1, \ldots, i_6) \). The set of issues \( T = \{ i_1, i_2, i_3, i_4, i_5, i_6 \} \) 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 \( \vec{x} \) found through an exhaustive search, an agent evaluates the utility on the basis of the utility function. If that utility is larger than the reservation value \( \delta \), 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 $T^M$ involving only the largest group including all issues first, and then 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 1 shows the tree-climbing protocol.

Algorithm 1 Tree-climbing protocol

$n$ agents with monotonic tree. ($u_1, \ldots, u_n$: Utility functions; $T$: 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 $T^0$ involving only the largest group including all issues first, and then 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.

Algorithm 2 Tree-descending protocol

$n$ agents with monotonic tree. ($u_1, \ldots, u_n$: Utility functions; $T$: 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

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:

- “(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.
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


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.