Compromising Strategies for Agents in Multiple Interdependent Issues Negotiation*

Shun OKUHARA^{†,††a)}, Student Member and Takayuki ITO^{††}, Member

SUMMARY This paper presents a compromising strategy based on constraint relaxation for automated negotiating agents in the nonlinear utility domain. Automated negotiating agents have been studied widely and are one of the key technologies for a future society in which multiple heterogeneous agents act collaboratively and competitively in order to help humans perform daily activities. A pressing issue is that most of the proposed negotiating agents utilize an ad-hoc compromising process, in which they basically just adjust/reduce a threshold to forcibly accept their opponents' offers. Because the threshold is just reduced and the agent just accepts the offer since the value is more than the threshold, it is very difficult to show how and what the agent conceded even after an agreement has been reached. To address this issue, we describe an explainable concession process using a constraint relaxation process. In this process, an agent changes its belief by relaxing constraints, i.e., removing constraints, so that it can accept it is the opponent's offer. We also propose three types of compromising strategies. Experimental results demonstrate that these strategies are efficient.

key words: automated negotiating agents, compromise, agreement

1. Introduction

Automated negotiating agents have been studied widely in the area of multi-agent systems [2]–[10], and heterogeneous, intelligent, and autonomous systems (agents) such as self-driving cars have been implemented in actual societies. In such implementations, conflicts may occur among multiple agents. A social mechanism that forces agents to reach an agreement to resolve conflicts through automated negotiation is thus required.

Many researchers are working on automated negotiating agents in the field of multi-agent systems, with international workshops and international competitions in this area being held since around 2010. Multi-agent systems are one of the most important technological advancements that have been made to address the needs of the next generation. The Automated Negotiating Agents Competition (ANAC) has been held since 2010 as a testbed for automatic negotiation agent research. The ANAC adopts a multi-issue utility model and an alternating-offer protocol. Many differ-

ent negotiating agents have been proposed because ANAC changes and extends the rules of negotiations every year. However, there are several drawbacks and problems that the ANAC competition cannot focus on. One of them is how to explain the compromise process. In negotiations, agents cannot reach an agreement if they consider only their own profits and interests. Therefore, the compromise strategy is essential to reach an agreement.

Most of the existing automated negotiating agents adopt *ad-hoc* compromising processes that only adjust their thresholds to accept the opponent's offer. This has made it difficult to explain how the compromise was achieved. It is important for automated negotiating agents in real society to explain how and why they compromised because they need to interact with actual human beings.

A constraint is a basic unit of utility. We define the utility space of an agent as a set of constraints that satisfy the issue values and the argument for them. When a constraint is satisfied, the agent gets a utility value for this constraint. For example, the issues involved in buying a car may include the car's color, price, and type. These issues are linked by certain constraints. Thus, there could be a constraint that says if the "type" is "sports car," then the "color" should be "red." Also, if the "type" is "sedan," then "color" is "white." Constraints generate values if they are satisfied, but do not generate any values if they are not satisfied.

In this paper, we assume *shared* issues and *individual* issues. In other words, we can say that agents agree if they have the same issue value for shared issues. For individual issues, each agent can choose issue values to make their utility as high as possible. An agent faces a trade-off between maximizing its own utility by satisfying the constraint as much as possible while simultaneously keeping the same share value as that of the opponent agent. In order to resolve this trade-off, agents perform a compromise.

In the compromise process for the strategy we propose, the agent removes constraints one by one from the set of its own constraints. Then, it tries to change the constraints' most preferable issue-value of the shared issue. If the agent can change the issue-value to one that is the same as the opponent's, they can reach an agreement. Removing constraints is called "constraint relaxation." Specifically, we assume that the agent has a believed constraint set (IN) and an un-believed (OUT) constraint set. In the initial state, it is assumed that all constraints are IN, and that in the constraint relaxation process, agents move certain constraints from IN to OUT. Various strategies are enabled when the

Manuscript received March 27, 2019.

Manuscript revised October 2, 2019.

Manuscript publicized January 21, 2020.

[†]The author is with Faculty of Medical Management and Information Science, School of Health Sciences, Fujita Health University, Toyoake-shi, 470–1192, Japan.

^{††}The authors are with Department of Computer Science, Nagoya Institute of Technology, Nagoya-shi, 466–8555, Japan.

*This paper is a comprehensive extension of the paper presented in PRICAI2019 [1].

a) E-mail: okuhara@itolab.nitech.ac.jp

DOI: 10.1587/transinf.2019IIP0011

agent moves constraints from IN to OUT.

The four methods we propose are:

- (1) Relaxation of constraints based on value
- (2) Random constraint relaxation
- (3) Constraint relaxation based on distance
- (4) Constraint relaxation based on value and distance

In Sect. 2 of this paper, we describe the automatic negotiation agent and negotiation protocol. In Sect. 3, we propose a compromise algorithm that is based on the newly proposed constraint relaxation. In Sect. 4, we describe and discuss experimental results. In Sect. 5, we clarify the difference between our methods and related research. We conclude in Sect. 6 with a brief summary and mention of future work.

2. Automated Negotiating Agents

2.1 Utility Hyper-Graph

An agent has a complex utility space [11]. A variety of representations have been proposed for complex utility spaces [12]–[14]. In the present work, we used hypergraph-based representations [15], [16] to focus on dependency between issues (nodes). A “hypergraph” is a mathematical representation in which an edge can join multiple nodes. A utility space using a hyper-graph is called a “utility hyper-graph,” where nodes are issues and edges are constraints. The utility space U of agent a is represented by hypergraph (I, C) , wherein $I_i \in I$ is an issues set (node) and C is a constraint set (edge). Each issue has an issue value (Issue Value) within a predetermined range D_i . For example, one issue (color) when purchasing a car has an issue value within a range of “red,” “blue,” and “green.” Constraint $C_j \in C$ is represented by $(v_{C_j}, \phi_{C_j}, \delta_{C_j})$, where v_{C_j} represents the value of constraint C_j and ϕ_{C_j} is a set of issues wherein constraint C_j is joined. Consequently, δ_{C_j} is a set of ranges where $\delta_{C_j} = \{range_{C_j}(I_i) : I_i \in \phi_{C_j}\}$. The conditions under which constraint C_j is satisfied are as follows. The value assumed by issues I_i is x_{I_i} . If C_j is satisfied, then an agent having C_j obtains the value thereof v_{C_j} .

$$C_j = \begin{cases} \text{satisfy} & \text{if } x_{I_i} \in range_{C_j}(I_i) \quad \forall I_i \in \phi_{C_j} \\ \text{unsatisfy} & \text{otherwise} \end{cases}$$

The properties of the hypergraph-based utility model are as follows.

- Nested network: The graph can be more complex, like a nested network, and the strategies proposed in this paper work even if there is a nested network.
- Independency of constraints: We assume that each constraint is independent, which means that when one constraint is relaxed, that constraint does not affect the other constraints. Also, in constraint relaxation, the relaxation of one node does not affect the utility of the other nodes.

Fig. 1 Shared issue and utility graph.

- Inter-dependency of issues: A constraint represents the inter-dependency of issues in negotiation. Thus, if we consider dependency between constraints, the model becomes more sophisticated but also more complex. Dependency between constraints will be a focus of our future work.
- Nonlinearity: The hypergraph representation has been proposed to represent non-linear utilities. That means it is not additive on alternatives. A constraint, which is the building block of our utility model, represents this nonlinearity on alternatives. Nonlinear utility functions have been studied extensively for treating more practical applications [11], [15], [17]–[19]. In general, nonlinear utility functions do not assume additivity.
- Applications: The hypergraph representation can deal with negotiations where participants have nonlinear utility functions. In general, people's utility functions are not linear. For example, in making the decision to buy a house, there are several issues and related alternatives/values. These issues and alternatives must be interdependent, and as such it is not additive on alternatives. The issues would be “place,” “price,” “style,” “neighborhoods,” etc. Alternatives on “place” would be “in city,” “near mountain,” “near river,” “near beach,” etc. Alternatives on “style” would be “modern,” “western,” “Japanese,” etc. If one chooses “near beach” as the “place,” then one would also choose “modern” as “style”. This is interdependency between issues and alternatives/values.

The hypergraph-based representation enables us to computationally handle these types of nonlinear utility functions, and it can be easily utilized for practical applications like decision support systems, as shown above.

Figure 1 shows an example of an agent's utility graph and issues shared.

Here, two agents who have their own separate utility graphs share three issues. Each of the agents has constraints that link these issues. Each issue takes an issue value. A constraint is satisfied if the issues linked by this constraint have issue values within the predefined ranges. When a constraint is satisfied, the agent obtains a value from this satisfied constraint.

Assumption 1: A constraint that is difficult to satisfy has a

higher value.

We made the following assumptions in accordance with assumption 1:

- Constraints with a wider issue-value range ($range_{C_j}$) are easier to satisfy and the values are lower. Constraints with a narrower value range are more difficult to satisfy and the values are higher.
- Agreement has higher priority, so constraints related to the shared issues have more value than individual constraints.

The search space is quite large. Therefore, it is difficult to design an efficient method to find non-conflict solutions that have higher utility. This is a key point in the field of automated negotiation agents.

Theoretically, the complexity to find a reasonable solution in a greedy way is as follows:

$$O(nm) \quad (1)$$

where n is the worst number of issues and m is the worst number of values in one issue. The worst number means that the total number of issues and values belongs to the accumulated constraints.

The conventional threshold-based approaches try to find a solution by incrementally reducing the threshold for each agent and then locating a solution that is both satisfied by various constraints and is above the threshold. In contrast, in our proposed method, we focus on the constraint first: specifically, we choose one constraint, remove it, and then find an agreement point that maximizes the utility without the removed constraint.

2.2 Negotiation Protocol

In order to focus only on the compromise algorithms, we propose a simple negotiation protocol: a simultaneous repeated offer protocol, where each agent proposes its own offer to the opponent. If both agents can accept the offers, they reach an agreement. If not, both agents revise their offers by compromising, and then propose again. This repeats until one of the agents cannot compromise anymore.

Algorithm 1 is the concrete definition.

By performing the compromise process, the agent modifies and revises its utility space so that it can compromise with the opponent to reach an agreement.

In this protocol, in each round, each agent makes an optimal proposal based on its own utility space. In the field of automated negotiations, the alternating of offered protocols [20] is well known and has been extensively utilized. However, an agent's strategy changes depending on which agent makes the first proposal. Therefore, we adopted a simple simultaneous repeated offer protocol. Extending it to alternating offer protocols is a subject for future work.

Algorithm 1 Simultaneous repeated offer protocol.

```

1: repeat
2:   Each agent finds an optimal issue value assignment that maximizes
   its own utility
3:   If there are multiple optimal assignments, each agent chooses one
   of them randomly.
4:   Each agent simultaneously proposes the issue value for the shared
   issue as an offer
5:   Judging agreement:
6:   if Both agents offer the same issue value for the shared issues then
7:     they reach an agreement
8:   else
9:     Each agent performs the compromise process (refer to the next
   section).
10:  end if
11: until one of the agents cannot continue, i.e., no constraint can be
   relaxed or when the prescribed number of iterations is reached.

```

3. Explainable Compromise Process Based on Constraint Relaxation

3.1 Explainable Compromise Process

In this section, we discuss the compromise process based on constraint relaxation. Constraint relaxation is the process of reducing the sum of allowable utilities (value) by reducing the number of satisfiable constraints.

Most previous works did not provide a mechanism to explain how the agreement is achieved on their threshold-based methods. In this research, satisfiable constraints are relaxed, i.e., removed, for compromising. Because we can ascertain which constraints are removed in the compromising process, we can understand how the agent compromised and which constraints were relaxed for agreement. In order to achieve this explainable compromising process, we classified the constraints into believed (IN) constraints and non-believed (OUT) constraints. Initially, all constraints are set to IN, while the relaxed constraints are set to OUT.

We provide the analysis of our main claim on the explainability of our protocol. Figures Fig. 2 and Fig. 3 show simple examples of the compromise process we propose. In Fig. 2, Agent 1 has Issue I_1 and Issue I_s . Issue I_s is the shared issue. Agent 2 has Issue I_2 and Issue I_s . Each issue has an issue value of 1, 2, or 3. Agent 1 has constraints C_1 and C_2 . Since the utility is higher when both issues are satisfied, the initial optimal solution is 1 for I_1 and 2 for I_s . Agent 2 has constraints C_3 , C_4 , and C_5 . The optimal solution is 3 for Issue I_s and 2 for I_2 . In this case, the agents have different issue values for the shared issue I_s , which means they have not reached an agreement. Therefore, each agent performs a compromise process by removing one constraint.

In Fig. 3, Agent 1 sets constraint C_1 to OUT and Agent 2 sets constraint C_5 to OUT. Consequently, Agent 1's I_s issue value stays at 2 while Agent 2's I_s issue value becomes 2. As a result, Agent 1 and Agent 2 reach an agreement. Since it is known which constraints were set to OUT (not believed) in the compromise, it becomes possible to explain

Fig. 2 Example of agreement based on constraint relaxation 1: Initialization.

Fig. 3 Example of agreement based on constraint relaxation 2: Agreement by relaxation.

which constraints were left out and why. This is different from existing studies in which agents simply adjust the threshold of acceptance.

As demonstrated in this example, it can be clearly understood which constraints are relaxed by each agent. Agent 1 removed constraint C_1 and Agent 2 removed constraint C_5 .

In practical application, users can understand which constraints should be relaxed for their agreement, as the above example shows. Conventional automated negotiation agents cannot provide an explanation like this because they do not assume any relaxation of the constraints in negotiation; they just set the threshold that forces agents to reduce the utility to be in agreement [2]–[10]. This explainability is a completely new approach in the field of agent negotiation.

Our protocol adheres to commonly held properties of a protocol of the kind where an agent can legally go back to its previous offer. Actually, in the field of automated negotiation in multi-agent systems, we usually do not assume that an agent can legally go back to its previous offer [2]–[10]. In our new protocol, actually, an agent can go back to its previous offer in the sense that the agent prefers the same offer, i.e., the same issue value, after some constraint relaxation. In our negotiation model, we separated shared issues from individual issues. The offer is actually for the shared issues, while the value choice in the individual issues affects changes to the offer. Thus, in our protocol, an agent can go back to the previous offer if the agent prefers it after

changing the values of individual issues by constraint relaxations. This is the novel point in our protocol compared to the conventional automated negotiation model.

The meaning/semantics of a constraint can be described as follows. A constraint is a small building block of the whole utility of an agent. We can see a constraint as a small portion of utility. An agent has many different constraints in its utility space. Each constraint has different issues, different issue values, and a different portion of utility. By accumulating these constraints, the agent can get the whole utility space. In the space, there could be two similar constraints that have the same issues but different issue values. Constraint relaxation is a way to find a compromise. An agent tries to find a compromise that can allow it to accept the opponent's offer. This compromise happens in a step-wise way rather than by turning off the possibility of accepting offers from the opponent.

3.2 Compromising Strategies

We propose the following four compromising strategies. All initial constraints are IN and all initial relaxed constraints are OUT.

random: One of the constraints in IN is randomly selected and pushed into OUT.

min: The lowest value constraint is selected from IN constraints and pushed into OUT.

distance: The constraint that has the longest distance from the shared issue is selected in IN and pushed into OUT. Here, the distance is the number of connected constraints from the shared issue.

min + distance: The constraint with the least value among the constraints most distant from the shared issue is selected from IN and pushed into OUT.

Algorithm 2 Distance: A greedy algorithm to obtain Distance.

```

1: Initialization
2:  $T := I_{shared}$ , where  $I_{shared}$  is a set of the shared issues.
3:  $distance := 0$ 
4: for  $c_j \in C$  do  $d_{c_j} := \phi_{C_j}(\phi_{C_j}$  is a set of issues wherein constraint  $C_i$  is joined.)
5: end for
6: while  $T \neq \emptyset$  do
7:    $I_i := car(T)$ 
8:   for each  $C_j$  in  $C$  do
9:     if  $I_i \in \phi_{C_j}$  and  $d_{C_j} = 0$  then
10:       $d_{C_j} := distance$ 
11:       $T := T + \phi_{C_j} - \{I_i\}$ 
12:    end if
13:     $distance = distance + 1$ 
14:   end for
15: end while

```

In the distance and min + distance strategies, we used to measure the distance between a constraint and a shared issue. Here, we use a simple greedy algorithm to calculate

the distance of a constraint from a shared issue.

Algorithm 2 shows the procedure in detail. It starts from one shared issue of the shared issues. The algorithm assigns the distance of 1 to the constraints that include this shared issue. Then, it gathers the other issues that are included in these constraints as a constraint set. The algorithm proceeds in the same manner for this constraint set. If the target set is empty, it stops. Lines 6 to 15 reflect this repeated assignment process. In line 7, the function “car(T)” means “get the first element of set T”.

4. Experiments

4.1 Experiment 1: Comparison with Baseline Strategy

(1) Experimental Setting

Due to the nature of the negotiation domain we defined in Sect. 2, it is not clear where we can provide formal proofs (e.g., mathematical or logical proofs). For this reason, we provide comprehensive experimental results to demonstrate the effectiveness of our methods *empirically*.

We performed an experiment to compare the performances of the proposed compromising strategies with the random strategy as the baseline. Our experimental setting included the following parameters:

- There are two agents.
- One issue can take up to ten values.
- There is a single shared issue.
- The number of constraints that include an issue is x .
- Each agent has y issues.
- Each constraint includes at least one issue.

In other words, each issue is always included in one or more constraints. We utilized a multi-start local search approach as the search method to find the optimal solution. Graph structures based on constraints and issues are assigned randomly. Our experimental setting implies a situation where there are a lot of issues and all of them are connected with a number of constraints.

We ran 1000 trials. In each trial, agents iterate at most N offers with the proposed simultaneous repeated offer protocol, where N is the number of constraints. The maximum number of iterations is equal to the number of constraints because an agent removes a single constraint in each iteration. In a single iteration, each agent optimizes its issue values by using the multiple start local search, where we set the value of 100 different restarts and 100 steps to search for each search. The values have been experimentally tuned.

(2) Results:

In this preliminary experiment, we ran trials for several different settings. The results obtained for three of these settings are shown in Fig. 4 to Fig. 12. The graphs compare the social welfare for min, random, distance, and distance+min when agents reached an agreement.

As shown in the figures, the social welfare results for

	the number
Number of Constraints Per Issue	$x=1$
Number of Issue Per Agent	$y=5$
Number of Constraints	5
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 4 Experimental results for $x = 1$ and $y = 5$.

	the number
Number of Constraints Per Issue	$x=1$
Number of Issue Per Agent	$y=10$
Number of Constraints	10
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 5 Experimental results for $x = 1$ and $y = 10$.

the min, distance, and distance+min categories were significantly higher than those for the random category. We use the Tukey's honest significance test for comparisons of all pairs of the samples. Each bar in the figures shows the number of samples as n . x is the number of constraints per issue and y is the number of issues per agent.

Figure 4 shows the experimental results for $x = 1$ and $y = 5$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 5 shows the experimental results for $x = 1$ and $y = 10$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant.

	the number
Number of Constraints Per Issue	$x=1$
Number of Issue Per Agent	$y=15$
Number of Constraints	15
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD
 *:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 6 Experimental results for $x = 1$ and $y = 15$.

	the number
Number of Constraints Per Issue	$x=1$
Number of Issue Per Agent	$y=25$
Number of Constraints	25
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD
 *:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 8 Experimental results for $x = 1$ and $y = 25$.

	the number
Number of Constraints Per Issue	$x=1$
Number of Issue Per Agent	$y=20$
Number of Constraints	20
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD
 *:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 7 Experimental results for $x = 1$ and $y = 20$.

	the number
Number of Constraints Per Issue	$x=2$
Number of Issue Per Agent	$y=5$
Number of Constraints	10
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD
 *:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 9 Experimental results for $x = 2$ and $y = 5$.

cally significant. Also, there is a significant difference between min and distance, indicating that in this case, the min strategy worked better than the distance strategy.

Figure 6 shows the experimental results for $x = 1$ and $y = 15$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 7 shows the experimental results for $x = 1$ and $y = 20$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 8 shows the experimental results for $x = 1$ and

$y = 25$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 9 shows the experimental results for $x = 2$ and $y = 5$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. Also, there is a significant difference between min and distance, indicating that in this case, the min strategy worked better than the distance strategy.

Figure 10 shows the experimental results for $x = 2$ and $y = 10$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among

	the number
Number of Constraints Per Issue	$x=2$
Number of Issue Per Agent	$y=10$
Number of Constraints	20
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 10 Experimental results for $x = 2$ and $y = 10$.

	the number
Number of Constraints Per Issue	$x=2$
Number of Issue Per Agent	$y=20$
Number of Constraints	40
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 12 Experimental results for $x = 2$ and $y = 20$.

	the number
Number of Constraints Per Issue	$x=2$
Number of Issue Per Agent	$y=15$
Number of Constraints	30
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 11 Experimental results for $x = 2$ and $y = 15$.

	the number
Number of Constraints Per Issue	$x=2$
Number of Issue Per Agent	$y=25$
Number of Constraints	50
Max Constraint Dimensions	3

comparisons for all pairs using Tukey-Kramer HSD

*: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 13 Experimental results for $x = 2$ and $y = 25$.

min, distance, and distance_min.

Figure 11 shows the experimental results for $x = 2$ and $y = 15$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 12 shows the experimental results for $x = 2$ and $y = 20$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

Figure 13 shows the experimental results for $x = 2$ and $y = 25$. As shown, min, distance, and distance_min had a higher social welfare compared with random. It is statistically significant. There is no significant difference among min, distance, and distance_min.

min, distance, and distance_min.

When there were more than 30 issues per agent, we were unable to obtain stable experiment results. Specifically, it was difficult to obtain results showing a significant difference in the drawing method. This is because when there were more than 30 issues per agent, the number of solutions exceeded 10^{30} , and considerable calculation was required to search for the optimal solution. Developing a scalable method will be an important focus of our future work. Also, the graph structure currently given to the agent is random. We intend to develop an optimization strategy based on the structure of the graph.

4.2 Experiment 2: Comparison with Threshold Adjustment

(1) Experimental setting:

We compared our compromising strategies with the conventional threshold adjustment strategy. Our compromising strategies include random, min, distance, and distance-min.

The parameters in our experimental setting are the same as in the first experiment:

- There are two agents.
- One issue can take up to ten values.
- There is a single shared issue.
- The number of constraints that include an issue is x .
- Each agent has y issues.
- Each constraint includes at least one issue.

We utilized a multi-start local search approach as the search method to find the optimal solution. Graph structures based on constraints and issues are assigned randomly. We ran 1000 trials. In each trial, agents iterate at most N offers with the proposed simultaneous repeated offer protocol, where N is the number of constraints. In a single iteration, each agent optimizes its issue values by using the multiple start local search, where we set the value of 100 different restarts and 100 steps to search for each search.

For the comparison, we designed a threshold adjustment algorithm (Algorithm 3) where, for each iteration, the agent decreases the threshold and tries to find an agreement point that exceeds that threshold. There have been many studies based on threshold adjustment algorithms, but here we use a simple but common one that has been widely recognized in the literature [2]–[10].

Algorithm 3 Threshold Adjustment Algorithm

```

1: Threshold number set as the maximum utility.  $Th = \max_{utility}$ 
2: repeat
3:   Each agent finds an issue value assignment that exceeds the threshold value  $Th$ .
4:   If there are multiple optimal assignments, each agent chooses one of them randomly.
5:   Each agent simultaneously proposes the issue value for the shared issue as an offer
6:   Judging agreement:
7:     if Both agents offer the same issue value for the shared issues then
8:       they reached an agreement
9:     else
10:      Each agent performs the compromise process (refer to the next section).
11:   end if
12:   Threshold is decreased at a certain number, here we assume it as 10.  $Th = Th - 10$ .
13: until one of the agents cannot continue, i.e., the threshold value of one agent is below 0.

```

(2) Results:

In this experiment, we ran trials for several different settings. The results are shown in Fig. 14 to Fig. 17. The

comparisons for all pairs using Tukey-Kramer HSD
*:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 14 Experimental result for concession degree $d = 1$.

comparisons for all pairs using Tukey-Kramer HSD
*:P < 0.05, statistically significant **:P < 0.01, statistically highly significant

Fig. 15 Experimental result for concession degree $d = 2$.

graphs compare the social welfare for min, random, distance, distance+min, and threshold when agents reached an agreement. In Fig. 14 to Fig. 17, the social welfare for the constraint relaxations including the random, min, distance, distance+min categories was significantly higher than that for the threshold adjustment method. We use the Tukey's honest significance test for comparisons of all pairs of the samples. Each bar in the figures shows the number of samples as n . We did experiments by changing the number of the concession degree from $d = 1$ to $d = 4$, as this number highly affects the performance of the concession. The concession degree means how much the agent removed the

comparisons for all pairs using Tukey-Kramer HSD
 *: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 16 Experimental result for concession degree $d = 3$.

comparisons for all pairs using Tukey-Kramer HSD
 *: $P < 0.05$, statistically significant **: $P < 0.01$, statistically highly significant

Fig. 17 Experimental result for concession degree $d = 4$.

constraints in one concession.

Figure 14 shows the experimental results for $d = 1$. The constraint relaxation strategies including min, distance, and distance_min had a higher social welfare compared with the threshold adjustment. It is statistically significant.

Figure 15 shows the experimental results for $d = 2$. Again, the constraint relaxation strategies including min, distance, and distance_min had a higher social welfare compared with the threshold adjustment. It is statistically significant as well.

Figure 16 shows the experimental results for $d = 3$. Again, the constraint relaxation strategies including min,

distance, and distance_min had a higher social welfare compared with the threshold adjustment. It is statistically significant as well.

Figure 17 shows the experimental results for $d = 4$. The constraint relaxation strategies including min, distance, and distance_min had a higher social welfare compared with the threshold adjustment. It is statistically significant.

As we have shown in Fig. 14 to Fig. 17, the constraint relaxation strategies could obtain a higher social welfare than the threshold adjustment strategy when the agents could reach an agreement.

5. Related Work and Comparison

5.1 Related Work

In this section, we describe the differences between our study and related work. In the field of automated negotiation research, the compromise process was first proposed by Klein et al. [21]. Their main argument is that it is reasonable for an agent to gradually compromise at the Pareto front in simple negotiations where the issues are independent and the utility space is linear in each issue. However, if the issues are interdependent, the process is not so simple because the utility space is complicated, which makes it harder for the agent to find the Pareto front. To address this problem, Klein et al. proposed an SA-based agreement point search protocol (implicitly assuming compromising). In addition, Faratin et al. [22] analyzed various compromise functions.

The ANAC Competition [23] has been held annually since 2010. It is common for ANAC agents to adopt a method for estimating and presenting proposals that can be statistically accepted from the opponent's offers and to accept the proposal by adjusting the threshold considering the time discount utility. For example, AgentK [24], the winning agent of ANAC 2010, estimates the opponent's utility space and the attitude (hostile or compromising) from the opponent's offer history to work towards agreement. If the partner seems to be a compromiser, a concession is made, and if the partner is hostile, it will not concede more than a certain threshold. This is the strategy that pioneered ANAC's basic concession strategy. Fawkes [25], the winning agent of ANAC 2013, estimates optimal concessions by using discrete wavelet prediction based on an opponent's offer history. Most existing studies have focused on how to adjust the threshold so that the opponent's offer can be accepted. The threshold is a kind of upper limitation utility with which the agent can accept the opponent's offer. However, these studies provide no explanation as to how to archive the threshold value. Thus, they do not explain why the agent compromises. This is a real problem because if your self-driving car compromises, you will not be able to obtain any explanation about the compromise. Also, to the best of our knowledge, there has been no research that assesses how compromising can be explained in an automated negotiation agent that assumes multi-argument utility func-

tions.

Sycara has published a series of studies [26]–[29] proposing negotiation and compromising processes that are explainable because they use case-based reasoning. Specifically, they define compromise and persuasion in the form of logical arguments within the framework of case-based reasoning. Sycara's series of studies is also related to argumentation theory [30], [31] and has been developed into mathematical argumentation theory. In contrast, our own research is focused on how to construct an explainable compromise process based on a utility function that can be handled numerically.

The Distributed Constraint Satisfaction/Optimization Problem (DCSP/DCOP) [32], [33] is one of the major topics in multi-agent research. Because our model is based on constraints, it is closely related to DCSP/DCOP. The main difference is that our model focuses on negotiation situations where agents are basically trying to maximize their own individual utilities, but they compromise because they need to make an agreement. This is because if they cannot reach an agreement, there is no utility. In DCSP/DCOP, however, agents basically do not consider their own individual utilities. Rather, the main focus is on constraint satisfaction or optimization with distributed cooperative agents. Wakaki et al. [30] published a paper about a Distributed Truth Maintenance System (DTMS) in which they proposed a classification of *consistency* in multi-agent environments. They classified the distributed consistency concept into Inconsistent, Local-Consistency, Local-and-Shared-Consistency, and Global Consistency categories. In this study, an agreement means that each agent has its internal consistency in addition to a consistent shared issue-value, which is the Local-and-Shared-Consistency. The compromising method we propose is one of the methods for obtaining Local-and-Shared-Consistency. However, the constraint graph we use represents utility space, while a DTMS does not express preferences.

5.2 Comparison between our Model and the Other Models

We have proposed a completely new negotiation model with constraint-graph-based utility. It is a totally different model from the conventional negotiation model with the threshold adjustment methods and the multi-attribute utility model. We clarify this difference in detail as follows.

In the conventional negotiation model, as far as we know, all negotiation models in automated negotiation agents [2]–[10] feature utility models that are based on the multi-attribute utility model. They do NOT consider graph features among constraints. Also, the negotiation models are essentially based on threshold adjustment methods [2]–[10]. While such methods are effective to achieve compromises, it is very difficult to explain why the compromising happened. In contrast, in our proposed negotiation model, we use the constraint-graph-based utility model [15], which represent an agent's utility as a graph consisting of constraints. A constraint represents a utility among several is-

sues. If each issue has a certain value, and if these issues are satisfied by a certain value, then this constraint is considered satisfied. If a constraint is satisfied, then an agent who has this constraint will obtain a utility. The accumulation of utilities of constraints represents the agent's utility. In this paper, we further extend the constraint-graph-utility model to include individual issues and shared issues among agents. Individual issues are local issues owned by an agent, while shared issues are shared among various agents. The concept of individual issues is a totally new idea, while shared issues are already common in the field of automated negotiating agents. In addition to the new constraint-graph utility model and individual/shared issues, our negotiation model, where each agent tries to remove a constraint for each negotiation step, is also completely new. This idea was inspired by the way humans compromise during negotiations in the real world. Our idea was to remove a constraint in order to compromise so as to make an agreement. This mirrors the kind of patience and endurance exhibited in the real world. In particular, in our negotiation model, in order to make an agreement about shared issues, agents try to compromise from their individual issues. In our model, agents can clearly show which constraints are removed in a certain order, which means our model is explainable. These are the elements that make our proposed negotiation model completely unique.

6. Conclusion

This paper proposed an explainable compromise process for automatic negotiation agents. Most existing automatic negotiation compromising processes are ad-hoc adjustments of a threshold to accept an opponent's offer.

In contrast, our proposed method enables an explanation to be obtained by eliminating the constraints one by one.

Our contributions are as follows. (1) The unique explainable compromise process we developed is based on a utility graph structured with constraints and issues. (2) For automatic multi-issue negotiation, we developed a new model that distinguishes between shared issues and personal issues. (3) For the compromise process, we developed a constraint relaxation process based on distance and value and demonstrated its effectiveness.

Experimental results demonstrate that methods (1), (3), and (4) are able to obtain social surpluses significantly higher than method (2).

The proposed method is a heuristic method that overcomes huge complexity to find a good agreement point in nonlinear utility-based negotiations. We proposed several negotiation strategies for this protocol. We also demonstrated the effectiveness of our proposed methods by means of empirical evidence in several experiments.

As a direction for future work, we will develop a more sophisticated compromising process, such as one that can find possible combinations of the fewest constraints to be relaxed so that agents can change their alternatives. The AI-

ternating Offer protocol [20] has long been one of the most popular protocols for bilateral negotiating agents. In this paper, we utilized a simple simultaneous offering protocol because our main objective was to develop a new compromising process for agents. Another possible avenue for future work is to extend our protocol to the alternating offers protocol.

In this paper, a constraint is independent from the other constraints in the sense that even if we remove one constraint, there is no effect to the other constraints. It is possible to think a different model in which if we remove one constraint, then this removal affects to the value of the other constraints. This is one of the other research directions.

Acknowledgments

This work was partially supported by the JST CREST fund (Grant Number: JPMJCR15E1).

References

- [1] S. Okuhara and T. Ito, "A compromising strategy based on constraint relaxation for automated negotiating agents," *Pacific Rim International Conference on Artificial Intelligence*, vol.11670, pp.675–687, Springer, 2019.
- [2] Q. Bai, F. Ren, K. Fujita, M. Zhang, and T. Ito, "Multi-agent and Complex Systems," vol.670, Springer, 2017.
- [3] N. Fukuta, T. Ito, M. Zhang, K. Fujita, and V. Robu, "Recent Advances in Agent-based Complex Automated Negotiation," vol.638, Springer, 2016.
- [4] K. Fujita, T. Ito, M. Zhang, and V. Robu, "Next Frontier in Agent-based Complex Automated Negotiation," vol.596, Springer, 2015.
- [5] I. Marsa-Maestre, M.A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai, and K. Fujita, "Novel insights in agent-based complex automated negotiation," vol.535, Springer, 2014.
- [6] T. Ito, M. Zhang, V. Robu, and T. Matsuo, "Complex automated negotiations: Theories, models, and software competitions," vol.435, Springer, 2013.
- [7] T. Ito, M. Zhang, V. Robu, S. Fatima, and T. Matsuo, "New trends in agent-based complex automated negotiations," vol.383, Springer, 2011.
- [8] T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki, "Innovations in Agent-Based Complex Automated Negotiations," vol.319, Springer, 2010.
- [9] T. Ito, M. Zhang, V. Robu, S. Fatima, and T. Matsuo, "Advances in agent-based complex automated negotiations," vol.233, Springer, 2009.
- [10] T. Ito, H. Hattori, M. Zhang, and T. Matsuo, "Rational, robust, and secure negotiations in multi-agent systems," vol.89, Springer, 2008.
- [11] T. Ito, H. Hattori, and M. Klein, "Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces," *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pp.1347–1352, 2007.
- [12] V. Robu, D.J.A. Somefun, and J.L. Poutré, "Modeling complex multi-issue negotiations using utility graphs," *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, pp.280–287, ACM, 2005.
- [13] V. Robu and H. La Poutré, "Constructing the structure of utility graphs used in multi-item negotiation through collaborative filtering of aggregate buyer preferences," *Rational, Robust, and Secure Negotiations in Multi-Agent Systems*, vol.89, pp.147–168, Springer, 2008.
- [14] R. Aydoğan, T. Baarslag, K.V. Hindriks, C.M. Jonker, and P. Yolum, "Heuristics for using cp-nets in utility-based negotiation without knowing utilities," *Knowledge and Information Systems*, vol.45, no.2, pp.357–388, 2015.
- [15] R. Hadfi and T. Ito, "On the complexity of utility hypergraphs," *Proceedings of the Seventh International Workshop on Agent-based Complex Automated Negotiations (ACAN2014)*, pp.89–105, 2014.
- [16] R. Hadfi and T. Ito, "Modeling complex nonlinear utility spaces using utility hyper-graphs," *Modeling Decisions for Artificial Intelligence*, vol.8825, pp.14–25, Springer, 2014.
- [17] H. Tamura and Y. Nakamura, "Decompositions of multiattribute utility functions based on convex dependence," *Operations Research*, vol.31, no.3, pp.488–506, 1983.
- [18] I. Marsa-Maestre, M.A. Lopez-Carmona, J.R. Velasco, and E. de la Hoz, "Effective bidding and deal identification for negotiations in highly nonlinear scenarios," *Proc. 8th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pp.1057–1064, 2009.
- [19] S.S. Fatima, M. Wooldridge, and N.R. Jennings, "An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions," *AAMAS (2)*, pp.1041–1048, 2009.
- [20] A. Rubinstein, "Perfect equilibrium in a bargaining model," *The Econometric Society*, vol.50, no.1, pp.97–109, 1982.
- [21] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam, "Negotiating complex contracts," *Group Decision and Negotiation*, vol.12, no.2, pp.58–73, 2003.
- [22] P. Faratin, C. Sierra, and N.R. Jennings, "Negotiation decision functions for autonomous agents," *Int. Journal of Robotics and Autonomous Systems*, vol.24, no.3–4, pp.159–182, 1998.
- [23] T. Baarslag, K. Fujita, E. Gerding, K. Hindriks, T. Ito, N.R. Jennings, C. Jonker, S. Kraus, R. Lin, V. Robu, and C. Williams, "The first international automated negotiating agents competition," *Artificial Intelligence Journal (AIJ)*, pp.113–135, 2012.
- [24] S. Kawaguchi, K. Fujita, and T. Ito, "Compromising strategy based on estimated maximum utility for automated negotiation agents competition (anac-10)," *Modern Approaches in Applied Intelligence*, vol.6704, pp.501–510, Springer, 2011.
- [25] T. Baarslag, "What to bid and when to stop," Master's thesis, Delft University of Technology, 2014.
- [26] K.P. Sycara, "Argumentation: Planning other agents' plans," *Proceedings on International Joint Conference on Artificial Intelligence (IJCAI-89)*, pp.517–523, 1989.
- [27] K.P. Sycara, "Argumentation: Planning other agents' plans," *Proceedings on International Joint Conference on Artificial Intelligence (IJCAI-89)*, pp.517–523, 1989.
- [28] K.P. Sycara, "Resolving goal conflicts via negotiation," *Proceedings of Fifth National Conference on Artificial Intelligence*, pp.245–250, 1988.
- [29] K. Sycara-Cyranski, "Arguments of persuasion in labor mediation," *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-85)*, pp.294–296, 1985.
- [30] T. Wakaki and K. Nitta, *Mathematical discussion*, Tokyo Denki University Press, 2017 (Japanese).
- [31] C. Sierra, N.R. Jennings, P. Noriega, and S. Parsons, "A framework for argumentation-based negotiation," *International Workshop on Agent Theories, Architectures, and Languages*, pp.177–192, 1997.
- [32] M. Yokoo, E.H. Durfee, T. Ishida, and K. Kuwabara, "The distributed constraint satisfaction problem: formalization and algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol.10, no.5, pp.673–685, Sept. 1998.
- [33] F. Fioretto, E. Pontelli, and W. Yeoh, "Distributed constraint optimization problems and applications: A survey," *Journal of Artificial Intelligence Research*, vol.61, pp.623–698, 2018.

Shun Okuhara received the B.S. in electrical and information engineering from Nagoya Institute of Technology, in 2009. He received the M.S. in Management of Technology from Nagoya Institute of Technology, in 2013. He expected a PhD in computer engineering from Nagoya Institute of Technology. Since 2013, He have been as an Assistant Professor in Fujita Health University.

Takayuki Ito is Professor of Nagoya Institute of Technology. He received the Doctor of Engineering from the Nagoya Institute of Technology in 2000. He was a JSPS research fellow, an associate professor of JAIST, and a visiting scholar at USC/ISI, Harvard University, and MIT twice. He was a board member of IFAAMAS, the PC-chair of AAMAS2013, PRIMA2009, General-Chair of PRIMA2014, IEEE ICA2016, is the Local Arrangements Chair of IJCAI2020, and was a SPC/PC member

in many top-level conferences (IJCAI, AAMAS, ECAI, AAAI, etc). He received the JSAI Achievement Award, the JSPS Prize, the Fundamental Research Award of JSSST, the Prize for Science and Technology of the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology (MEXT), the Young Scientists' Prize of the Commendation for Science and Technology by the MEXT, the Nagao Special Research Award of IPSJ, the Best Paper Award of AAMAS2006, the 2005 Best Paper Award of JSSST, and the Super Creator Award of 2004 IPA Exploratory Software Creation Project. He was a JST PREST Researcher, and a principal investigator of the Japan Cabinet Funding Program for Next Generation World-Leading Researchers. He is currently principal investigator of JST CREST project.