PAPER

Weighting Estimation Methods for Opponents' Utility Functions Using Boosting in Multi-Time Negotiations[*]

Takaki MATSUNE[†a)], *Nonmember and* **Katsuhide FUJITA**[†b)], *Member*

SUMMARY Recently, multi-issue closed negotiations have attracted attention in multi-agent systems. In particular, multi-time and multilateral negotiation strategies are important topics in multi-issue closed negotiations. In multi-issue closed negotiations, an automated negotiating agent needs to have strategies for estimating an opponent's utility function by learning the opponent's behaviors since the opponent's utility information is not open to others. However, it is difficult to estimate an opponent's utility function for the following reasons: (1) Training datasets for estimating opponents' utility functions cannot be obtained. (2) It is difficult to apply the learned model to different negotiation domains and opponents. In this paper, we propose a novel method of estimating the opponents' utility functions using boosting based on the least-squares method and nonlinear programming. Our proposed method weights each utility function estimated by several existing utility function estimation methods and outputs improved utility function by summing each weighted function. The existing methods using boosting are based on the frequency-based method, which counts the number of values offered, considering the time elapsed when they offered. Our experimental results demonstrate that the accuracy of estimating opponents' utility functions is significantly improved under various conditions compared with the existing utility function estimation methods without boosting.

key words: *automated multi-issue negotiation, automated negotiating agents competition, multi-time negotiation, multilateral negotiation, boosting*

1. Introduction

In multi-agent system research, automated negotiation is playing an important role ([2]–[5]). Achievement of the automated negotiating agent will enable several independent and autonomous agents to automatically negotiate and act cooperatively in the case of conflict among them. Additionally, development of automated negotiating agents for realistic situations has the potential to support negotiations among people and achieve decision-support systems.

The latest Automated Negotiating Agents Competition (ANAC) [6] focused on the three-party multi-issue closed negotiation problem [7], assuming a case where negotiation is conducted by three or more agents. It also considered the realistic negotiation model, such as the discount factor and adoption of realistic negotiation scenarios. In addition, multi-time negotiations, which can utilize past negotiation records by repeating negotiations with the same agent in the same domain, have also attracted attention. By saving bid information exchanged in the past and utilizing the saved information in the next negotiations with the same domain, an agent can estimate the opponent's utility function more accurately by using machine learning. For example, when the opponent makes the same offer many times, it is likely to be an effective bid for the opponent. On the other hand, the bid in the final stage for making agreements may indicate the lowest utility to make agreements for the opponent.

To proceed negotiations with advantage, it is important to learn an opponent's negotiation strategy appropriately and to utilize the learned model for one's own negotiation strategy. For these purposes, a method was proposed of predicting how the opponent will compromise in the future as time elapses by evaluating the bids proposed by the opponent using TrAdaboost [8]. Despite some existing methods for estimating an opponent's utility function, further improvement of estimation accuracy is still important [9].

This paper proposes a novel method to improve the estimation accuracy, assuming multilateral multi-time closed negotiations. By estimating opponent utility function accurately, it can apply to negotiation strategies. For example, the applied strategy can offer or reject the opponents' offers to obtain the higher individual utility. In addition, the agent with our proposed approach can estimate a Pareto front, which is the one of the optimal agreements in the negotiation, and it also can minimize the effort of the discount factor by reaching an agreement at an early stage.

Our proposed method combines several existing utility function estimation methods using boosting [10] based on the least-squares method and nonlinear programming. The existing methods to be used with boosting are based on the frequency-based method, which counts the number of values offered, considering the time elapsed when they offer bids [11]. Utility functions that are suitable for opponents and domains can be estimated appropriately as our proposed method can combine several utility function estimation methods by adding weights and summing them up appropriately using boosting. In addition to this, we propose a method to estimate a Pareto front using the estimated opponents' utility function.

Experiments evaluating our proposed utility function estimation method are conducted. We demonstrate that our proposed method can estimate more accurately compared with the existing methods in many cases.

The rest of the paper is organized as follows. First, we

Manuscript received February 7, 2018.
Manuscript revised June 2, 2018.
Manuscript publicized July 10, 2018.

[†]The authors are with Department of Computer and Information Sciences, Graduate School of Engineering, Tokyo University of Agriculture and Technology, Tokyo, 184–8588 Japan.

[*]This paper is an extended version of work published in [1].

a) E-mail: matsune@katfuji.lab.tuat.ac.jp
b) E-mail: katfuji@cc.tuat.ac.jp

DOI: 10.1587/transinf.2018EDP7056

describe the multi-issue closed negotiation problem. Next, we propose an estimation method of opponents' utility functions using boosting. Then, the proposed utility function estimation methods will be evaluated. Finally, we present our conclusions.

2. Related Works

This paper focuses on research in the area of multi-issue closed negotiation, which is an important class of real-life negotiations. Closed negotiation means that opponents do not reveal their preferences to each other. Negotiating agents designed using a heuristic approach require extensive evaluation, typically through simulations and empirical analysis, since it is usually impossible to predict precisely how the system and the constituent agents will behave in a wide variety of circumstances. Motivated by the challenges of bilateral negotiations between people and automated agents, the Automated Negotiating Agents Competition (ANAC) was organized in 2010 [12] to facilitate research in the area of multi-issue closed negotiation.

The followings are the competition's declared goals: (1) to encourage the development of practical negotiation agents that can proficiently negotiate against unknown opponents in a variety of circumstances; (2) to provide a benchmark for objectively evaluating different negotiation strategies; (3) to explore different learning and adaptation strategies and opponent models; (4) to collect state-of-the-art negotiating agents and scenarios and make them available to the wider research community. The competition was based on the Genius environment: the General Environment for Negotiation with Intelligent multi-purpose Usage Simulation [13].

By analyzing the ANAC results, the stream of the ANAC strategies and the important factors for developing the competition have been shown. Baarslag et al. presented an in-depth analysis and the key insights gained from ANAC 2011 [14]. This paper mainly analyzed different strategies using the classifications of agents with respect to their concession behavior against a set of standard benchmark strategies and empirical game theory (EGT) to investigate their robustness. It also showed that even though the most adaptive negotiation strategies are robust across different opponents, they are not necessarily the ones that win competitions. Furthermore, the EGT analysis highlights the importance of considering metrics.

Chen and Weiss proposed a negotiation approach called OMAC, which learns an opponent's strategy to predict the future utilities of counteroffers by discrete wavelet decomposition and cubic smoothing splines [15]. They also presented a negotiation strategy called EMAR for such environments that relies on a combination of Empirical Mode Decomposition (EMD) and Autoregressive Moving Average (ARMA) [16]. EMAR enables a negotiating agent to acquire an opponent model and to use it to adjust its target utility in real time on the basis of an adaptive concession-making mechanism.

Williams et al. proposed a novel negotiating agent based on Gaussian processes in multi-issue automated negotiation against unknown opponents [17]. Baarslag et al. focused on the acceptance dilemma; accepting the current offer may be suboptimal, since better offers might still be presented [18].

These above strategies mainly focus on bilateral negotiations. However, in recent years, multilateral negotiations have also attracted attention. From ANAC 2015, a multi-lateral negotiation protocol [19] was used instead of bilateral negotiation protocol [20] because a multi-lateral negotiation assumes more realistic scenarios. After that, many negotiation strategies have been proposed ([21]–[24]). In addition, Shinohara et al. proposed a negotiation protocol, which considers fair privacy for multilateral closed negotiation [25]. This protocol adaptively changes the proposal order to adjust the fairness of the revealed private information. While our proposed method can be used to bilateral negotiation problems, this paper focuses on multilateral, multi-time negotiation mainly because we can demonstrate that our approach works appropriately in state-of-the-art (ANAC2016) negotiating agents.

3. Multi-Issue Closed Negotiation

This paper considers the multilateral multi-issue closed negotiation problem. A negotiation problem (domain) D includes n issues $I_1, \ldots, I_n$ and each issue I_i includes k values $v_1^i, \ldots, v_k^i$. To v_c^i, the value of issue I_i, an evaluation value in integral number $eval(v_c^i) \in [0, +\infty)$ is set, and an issue weight $w_i \in \mathbb{R}$ is given to each issue. The issue weight w_i satisfies $\Sigma_{i=1}^n w_i = 1$ as well as $w_i \geq 0$. A bid is shown as $\vec{b} = [v_c^1, \ldots, v_c^n]$, assuming that a value c is selected for each issue I_i, and utility function $U(\vec{b})$ that determines the utility of the agent is shown as formula (1). Each agent has a different utility function, respectively.

$$U(\vec{b}) = \sum_{i=1}^{n} w_i \times \frac{eval(v_c^i)}{\max(eval(I_i))} \tag{1}$$

In this paper, the normalized time in the range of $t \in [0, 1]$ is used. $t = 0$ means the starting time of the negotiation, and $t = 1$ means that the negotiation was past the deadline. Some domains have discount factors (DF) and reservation values (RV) [26]. As the discount factor d, the actual utility that can be actually acquired is reduced according to time t as shown in formula (2). The reservation value is the minimum utility acquired in this domain where agreement is not achieved, and the discount factor is applied to the reservation value.

$$U_D(\vec{b}) = U(\vec{b}) \times d^t \tag{2}$$

In this paper, we consider Stacked Alternating Offers Protocol (SAOP) [19], which extends the application of Alternative Offers [20] from bilateral negotiations to three-party or more cases. Flow of SAOP is shown in Fig. 1. We consider the case in which Agents A, B, and C conduct a negotiation. In SAOP, each agent takes action always in the order

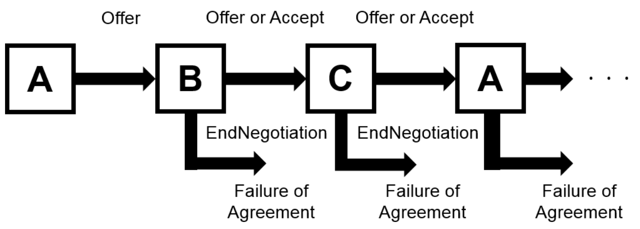


Fig. 1 SAOP flow chart.

of agent A, agent B, and agent C. At the beginning, agent A proposes a bid to agent B. Next, agent B takes one of the following actions to agent C.

- Offer: Rejecting the previous bid and proposing a new bid.
- Accept: Accepting the previous bid.
- EndNegotiation: Ending the negotiation without making an agreement before the deadline reaches. This action can be used to prevent losing a utility when Reservation Value (RV) and Discount Factor (DF) are set.

Afterwards, agent C takes a new action to agent A, and each agent will proceed in turn. In SAOP, negotiations will be continued until the following conditions are satisfied.

- All agents accept the bid, except for the one who offered it.
- Negotiation deadline passes before agreement was reached.
- One of the agents terminates the negotiation by selecting EndNegotiation.

The negotiation is successful only when all agents have accepted the bid except for the one who offered it. All actions by the agents are shared by all agents. When the negotiations are conducted several times under the same opponent and the same profile, the bid and the time of the proposal can be taken over. Therefore, the opponent's strategy and the opponents' utility functions can be estimated by machine learning and so on.

4. Estimation of Opponent's Utility Functions Using Boosting

We propose a novel method of estimating opponents' utility functions using boosting based on the least-squares method and nonlinear programming. In addition, we propose a method to estimate a Pareto front using the estimated opponents' utility function. The proposed method focuses on negotiations that are repeated several times for the same opponent under the same domains.

4.1 Frequency-Based Estimation Method of Opponent's Utility Function

Ikarashi et al. proposed a method of estimating opponents' utility function by counting the number of values offered, considering the elapsed rounds [11]. However, the relationship of the actual elapsed time and rounds elapsed is affected by the machine performance. Therefore, we modified the estimation method to use the normalized time in the range of [0, 1] as variables for weight addition, not the number of rounds as used in the existing works:

(1) Unweighted:
$V_i = \sum_{k=1}^{n} \mathit{Offered}(i,k)$

(2) Upward-sloping linear function:
$V_i = \sum_{k=1}^{n} t_k \cdot \mathit{Offered}(i,k)$

(3) Downward-sloping linear function:
$V_i = \sum_{k=1}^{n} (1 - t_k) \cdot \mathit{Offered}(i,k)$

(4) Convex downward quadratic function:
$V_i = \sum_{k=1}^{n} (t_k - \frac{1}{2})^2 \cdot \mathit{Offered}(i,k)$

(5) Convex upward quadratic function:
$V_i = \sum_{k=1}^{n} \{-(t_k - \frac{1}{2})^2 + \frac{1}{4}\} \cdot \mathit{Offered}(i,k)$

(6) Exponential upward-sloping function:
$V_i = \sum_{k=1}^{n} \mathrm{e}^{\frac{t_k}{\xi}} \cdot \mathit{Offered}(i,k)$

V_i is the estimated evaluation value for the ith value of a single issue. t_k means the time of the kth proposal by the opponent. $\mathit{Offered}(i,k)$ is the binary function that returns 1 when the ith value among $v_1,\dots,v_m$ was proposed, and 0 otherwise. m is the number of values in the domain. Although the weighting function based on logarithmic function was also included in the existing works, it is omitted in this paper. This is because that a logarithm function $\log t$ returns a negative value when t is $0 < t < 1$. Therefore, we cannot apply a logarithm function to our approach directly. In addition, the experimental result of Ikarashi et al. [11] shows that logarithm function was not so effective.

4.2 Estimation Method of Opponent's Utility Function Using Boosting

We propose a novel method of estimating an opponent's utility function by weighting the existing utility function estimation methods (called learner) and summing each weighted learner. This method only works when the negotiations under the same opponents and domains are repeated.

Initialization

Against N learners, the weighting vector of the learner $\mathcal{W} = \{w_1,\dots,w_{N+1}\}$ is prepared. The value of each element is initialized as $\frac{1}{N+1}$. The vector $\mathcal{U}^{(R)} = \{u_1^{(R)},\dots,u_N^{(R)},u_{N+1}^{(R)}\}$ for saving the utility calculated by the learners is prepared (R is the number of opponent's offers in a negotiation).

Step 1

The estimated utility of offered bids by each learner $\mathcal{L} = \{L_1,\dots,L_N,L_{N+1}\}$ which contains estimated evaluation values for the domain $L = \{V_1,\dots,V_m\}$ is calculated. m

is the number of values in the domain. $\mathcal{L}$ is added to $\mathcal{U}^{(R)}$ respectively as $\{u_1^{(r)}, \ldots, u_N^{(r)}, u_{N+1}^{(r)}\}$ (r is the number of received bids from the beginning of the negotiation to now). The learner L_{N+1} is a bias that outputs all value evaluations as 1. This bias has the following roles in the process of boosting:

(1) **Prevention of selecting specific learners only:** The weight is unevenly added to one learner in some cases without the bias when the output curves of all learners are lower than the compromising model curve $C(t)$ in nonlinear programming in Step 2.

(2) **Correction to the value with low evaluation:** The utility function estimation methods based on frequencies of bids have a tendency to consider the value including a small number of proposals as an extremely low score. By adding the bias, the extremely low value evaluation can be corrected.

Step 2

Solve formula (3) using nonlinear programming and set the solution as learner's weight ($\mathcal{W} = \{w_1, \ldots, w_{N+1}\}$).

$$\begin{aligned} &\mathit{minimize} \textstyle\sum_{i=1}^{r} (\sum_{j=1}^{N+1} (u_j^{(i)} \cdot w_j) - C(t))^2 \\ &s.t. \textstyle\sum_{j=1}^{N+1} w_j = 1 \\ &w_j \geq 0 \; (j = 1, \ldots, N+1) \\ &w_{N+1} \leq \tfrac{1}{N+1} \end{aligned} \tag{3}$$

The first line of formula (3) is a squared error between $u_j^{(i)}$, which is the utility estimated using the learner L_j weighted by w_j and the utility by the compromising model curve $C(t)$ in the time t. The compromising model curve $C(t)$ means the function of time t that predicts how the utility of the opponent's bids will compromise its own utility in the future. Most of the agents' strategies try to compromise as time elapses. Therefore, we consider the compromising model curve as the function in which the utility $[0, 1]$ will be reduced as time t increases. We set the maximum w_{N+1} as $\frac{1}{N+1}$ to prevent it from being too large.

Step 3

The output learner L_w is obtained by summing each weighted learner as formula (4).

$$L_w = \sum_{i=1}^{N+1} L_i \cdot w_i \tag{4}$$

4.3 Estimating Pareto Fronts Using Estimated Opponent's Utility Function

We also propose a method that estimates Pareto fronts using the estimated opponent's utility function between two agents.

(1) The list of all possible bids is generated within the domain.

(2) The utility of the listed bids is estimated using the estimated opponent's utility function.

(3) Bids with identical utility on opponent's utility functions are removed from the list except for the one with the highest utility for the opponent.

(4) The list is sorted in ascending order based on one's own utility.

(5) From the bids with the lowest utility, other bids are evaluated to determine whether the utilities of the opponent are smaller based on the sorted list. When the evaluated bid is smaller for the opponent's utility, the bid is identified as a Pareto optimal bid.

Based on the above procedures, our method can estimate Pareto fronts based on the estimated opponent's utility function. In the proposal and acceptance strategies in a negotiation, one important measure is Pareto optimality. If our proposed method can also estimate Pareto optimal bids, the agent's strategy will be very effective.

5. Experiments

5.1 Determining Parameters $C(t)$ for Our Proposed Method

In our hypothesis, when $C(t)$ is exactly the same as the utility of the bid proposed by the agents, the estimation accuracy is considered to be the best. However, the agents cannot obtain the real utility for the bid proposed by opponents in the negotiation because this paper assumes closed negotiation problems.

Before we decide the parameter $C(t)$, we recorded the negotiation histories with the same settings of Sect. 5.2 and analyzed the behavior of ANAC2016 agents. The experimental results demonstrate that the utility of a bid proposed by the opponent varies greatly depending on opponents' strategy (Fig. 2). Especially, some agents do not make

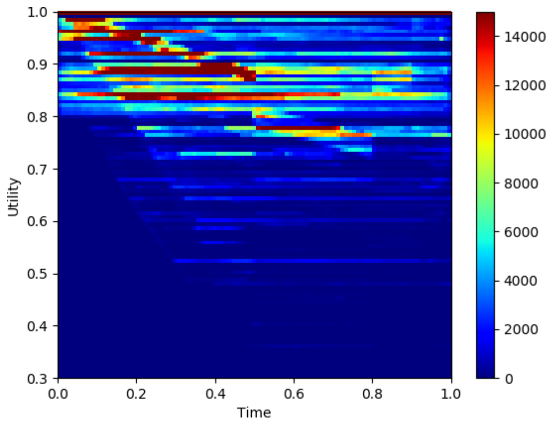


Fig. 2 The histogram of the time elapsed and utility of proposed bid. Approximately 30 million offers have been recorded in 840 negotiations.

any concessions until the time reaches near the negotiation deadline. On the other hands, the most agents with the concession strategies follow in the range of two simple models ($1.0 - 0.2t$ and $1.0 - 0.3t$). For these reasons, we decided to use these simple models: $1.0 - 0.2t$ and $1.0 - 0.3t$ by focusing on the changes of the estimation accuracies considering the behavior of state-of-the-art agents with the concession strategy.

5.2 Estimation Errors in ANAC2016 Domains

5.2.1 Experimental Settings

In experiments that estimated opponents' utility functions, we used the Genius [13] negotiation platform, which was developed to facilitate research in multi-issue negotiations. For learning opponents' utility functions and domains, the negotiation history is recorded by round-robin tournaments with the domains and agents used in ANAC2016. The following are the details of the experimental settings:

- Tournament setting: Round-robin.
- Agents: Top seven agents in individual utility category in ANAC2016: Caduceus, YXAgent, ParsCat, Farma, MyAgent, Atlas3, and Ngent.
- Domains: Four types of domains in ANAC2016 (Table 1). The profiles of each agent are respectively used from 1 to 3 in each domain.
- Number of agents per negotiation: 3.
- Negotiation time: 180 seconds.
- Total number of negotiations: $210 \times 4 = 840$.

The following are the settings of the estimation method in the experiments:

- The parameter of the exponential function was set to $\xi = 0.3$.
- Because the sizes of the obtained estimated utilities are different depending on the weighting function, all of the estimated utilities were normalized as $[0, 1]$. In the normalization, the estimated utilities were divided by the maximum utility in each issue where the target value belongs.
- Only the value in each issue was evaluated since no estimation was conducted for the weights of each issue.
- *(Estimation error) = abs((Estimated utility of each value) − (Correct utility of each value)).*

In boosting our proposed method, we evaluated the weights just before finishing the negotiation because the weights of each estimated utility function are updated sequentially.

Table 1 Domains used in experiments.

Domain	Issues	Domain Size	DF	RV
Clockwork	3	120	0.6	0.4
MyAgent	5	1280	0.7	0.2
MaxOops	7	7200	1.0	0.0
Terra	7	35840	0.5	0.4

We compared ten methods: six existing methods described in Sect. 4.1 above and four proposed methods using two kinds of weighting functions and two compromising models:

- **Unweighted:** Counting the number of bids without weighting
- **Upward-sloping linear function:** Upward-sloping linear function
- **Downward-sloping linear function:** Downward-sloping linear function
- **Downward-convex quadratic function:** Downward-convex quadratic function
- **Upward-convex quadratic function:** Upward-convex quadratic function
- **Exponential function:** Exponential function
- **Proposed method (1):** these weighting functions were used: Unweighted, Upward-sloping linear, Downward-sloping linear, Downward-convex quadratic, Convex quadratic, and Exponential functions. The compromising model curve formula is $C(t) = 1.0 - 0.2t$.
- **Proposed method (2):** these weighting functions were used: Unweighted, Upward-sloping linear, Downward-sloping linear, Downward-convex quadratic, Convex quadratic, and Exponential functions. The compromising model curve formula is $C(t) = 1.0 - 0.3t$.
- **Proposed method (3):** these weighting functions were used: Unweighted, Downward-sloping linear function, and Downward-convex quadratic functions. The compromising model curve formula is $C(t) = 1.0 - 0.2t$.
- **Proposed method (4):** these weighting functions were used: Unweighted, Downward-sloping linear function, and Downward-convex quadratic functions. The compromising model curve formula is $C(t) = 1.0 - 0.3t$.

5.2.2 Experimental Results

Figure 3 shows the average estimation errors in all the domains. Tables 2–5 show the details of the estimation errors for each domain. The average estimation error of the proposed method is reduced more than the existing methods. Since the estimation errors of proposed methods (3) and (4) are lower than proposed methods (1) and (2), two of our proposed methods effectively estimated opponent utilities by accurately weighting all learners during boosting. We confirmed that our proposed method reduced the estimation error to about 59% in the Clockwork domain (Table 2). In the MaxOops domain, however, it only reduced the estimation error to about 82% (Table 3).

Considering the YXAgent result in Table 3, we confirmed that cases exist where the result is worse than using the existing method. This is because the bias used in our proposed method was weighted inappropriately. In addition, we confirmed cases where the estimated utility became too high by excessively weighting the learners. Therefore, improvements of the limitations of weighting the learners are necessary. In addition, the minor differences between the

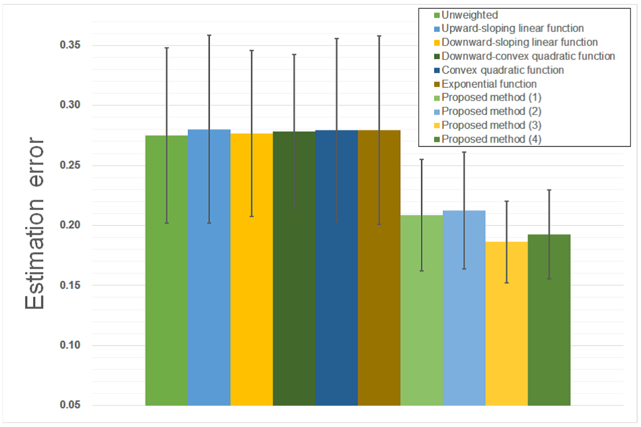


Fig. 3 Estimation error averages in all domains[†,††].

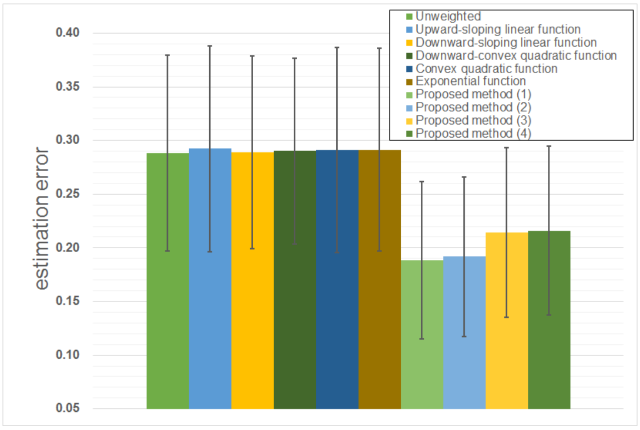


Fig. 4 Estimation error averages in all domains in bilateral negotiation[†,††].

compromising models are small because both compromising models are simple linear functions and do not substantially affect the results of the formula (3).

Our proposed method improved the average estimation errors because it can optimize the combinations of many frequency-based estimation methods. In other words, the purpose of this paper can be demonstrated in ANAC2016 domains. However, our proposed method is worsened when estimation methods with low performances are investigated in boosting. This is because that it addresses the learners as much as possible even though the performances of estimating opponents' utility functions might be worsened.

5.3 Estimation Errors in Bilateral Negotiations

5.3.1 Experimental Settings

Although this paper focuses on multilateral negotiation, our proposed method can also apply to bilateral negotiations. In this section, we conducted experiments to show the accuracy of our method in bilateral negotiations. The experimental settings are the same as in Sect. 5.2, except as follows:

- Number of agents per negotiation: 2.
- Negotiation protocol: SAOP. However, this protocol works substantially equivalent to Alternating Offers Protocol in bilateral negotiations.
- Total number of negotiations: $126 \times 4 = 504$.

5.3.2 Experimental Results

Figure 4 shows the average estimation errors in all the domains. Comparing the result in multilateral negotiations (Fig. 3) with the one in bilateral negotiations (Fig. 4), the proposed methods outperform the conventional methods in the both cases. Therefore, although the method with the highest accuracy is different in bilateral and multilateral negotiations, our proposed method can perform well in both cases.

[†]The error bars denote the standard deviation of the results.
[††]Experimental results have high standard deviations and are not statistically significant because the results are not shown for each domain or agent.

5.4 Estimation Errors in Changing Discount Factors

5.4.1 Experimental Settings

In the previous section, we demonstrated that our proposed methods can accurately estimate the utility functions of opponents in ANAC2016 domains. In general, most agent strategies propose various bids when the discount factor is small to reach a compromise at an earlier stage. It can also improve our method's accuracy because it prevents specific values from being estimated as high values by proposing them many times before compromising.

Experiments were conducted to demonstrate the estimation errors in changing discount factors. Experimental settings are the same as in Sect. 5.2, except as follows:

- Domains: party_domain (included in GENIUS). The profiles of each agent are respectively used from 1 to 3 in each domain.
- Negotiation time: 30 seconds.
- Discount factors (DF): 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1
- Total number of negotiations: $210 \times 10 = 2100$.
- The proposed method (3) was used for our estimation method.

5.4.2 Experimental Results

Tables 6 and 7 show the average estimation errors when the discount factors are changed.

Comparing both tables, since our proposed method outperforms the simple unweighted frequency-based method in various discount factors, it can estimate opponents' utility functions by combining some simple frequency-based estimation methods.

Table 2 Details of estimation error averages (Domain: Clockwork)[†††].

Weighting Function	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
Unweighted	0.29244	0.34910	0.18790	0.22072	0.34257	0.35582	0.25493	0.33883
Upward-sloping linear function	0.29344	0.34214	0.18774	0.22321	0.37446	0.35214	0.22774	0.34923
Downward-sloping linear function	0.29384	0.35287	0.18799	0.23532	0.32949	0.35618	0.26058	0.33719
Downward-convex quadratic function	0.30445	0.35352	0.18793	0.27712	0.33483	0.35665	0.28074	0.34269
Upward-convex quadratic function	0.29333	0.34719	0.18783	0.22840	0.36062	0.35303	0.22992	0.34899
Exponential function	0.29351	0.33968	0.18778	0.21336	0.37358	0.35412	0.24134	0.34740
Proposed method (1)	0.19859	0.23564	0.12631	0.13527	0.23001	0.28029	0.14898	0.23727
Proposed method (2)	0.19883	0.23597	0.12634	0.14004	0.22980	0.28029	0.14868	0.23434
Proposed method (3)	**0.17135**	**0.19593**	0.11295	**0.12307**	**0.19600**	**0.24170**	**0.13521**	0.19770
Proposed method (4)	0.17150	0.19630	**0.11283**	0.12349	0.19825	0.24170	0.13522	**0.19583**

Table 3 Details of estimation error averages (Domain: MaxOops)[†††].

Weighting Function	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
Unweighted	0.26181	0.33040	0.17735	0.10969	0.34594	0.14313	0.30823	0.41790
Upward-sloping linear function	0.26823	0.32827	0.18015	0.13627	0.35971	0.13843	0.30884	0.42598
Downward-sloping linear function	0.26485	0.33851	0.18605	0.10962	0.34819	0.14737	0.31192	0.41228
Downward-convex quadratic function	0.26690	0.33759	0.19320	0.12178	0.34117	0.15415	0.31242	0.40799
Upward-convex quadratic function	0.26580	0.32954	0.18008	0.12634	0.35192	0.14073	0.30825	0.42373
Exponential function	0.26893	0.32971	0.18032	0.13470	0.36389	0.13865	0.30898	0.42626
Proposed method (1)	0.23092	0.26629	**0.18227**	0.12486	0.29123	0.12008	0.28303	0.34867
Proposed method (2)	0.23842	0.28217	0.18485	0.11215	0.30680	0.13237	0.30115	0.34945
Proposed method (3)	**0.21342**	**0.23752**	0.19039	0.11215	**0.26142**	**0.11201**	**0.26817**	**0.31227**
Proposed method (4)	0.22542	0.25873	0.18959	**0.10694**	0.28555	0.12896	0.29524	0.31293

Table 4 Details of estimation error averages (Domain: MyAgent)[†††].

Weighting Function	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
Unweighted	0.29465	0.29728	0.27436	0.22195	0.34324	0.24649	0.31832	0.36090
Upward-sloping linear function	0.30220	0.29347	0.27360	0.23406	0.36040	0.24434	0.33978	0.36974
Downward-sloping linear function	0.29597	0.30458	0.27392	0.23416	0.33911	0.24647	0.31288	0.36067
Downward-convex quadratic function	0.29340	0.29921	0.27327	0.23271	0.34450	0.24940	0.30860	0.34607
Upward-convex quadratic function	0.30160	0.29688	0.27369	0.23604	0.35252	0.24353	0.33470	0.37384
Exponential function	0.30099	0.29422	0.27403	0.22678	0.36803	0.24697	0.33389	0.36302
Proposed method (1)	0.23187	0.20846	0.25824	0.19633	0.24826	0.21823	0.24085	0.25270
Proposed method (2)	0.23655	0.21658	0.26294	**0.19393**	0.26533	0.22037	0.24389	0.25280
Proposed method (3)	**0.21357**	**0.18486**	**0.25533**	0.19617	**0.22194**	**0.21486**	**0.20326**	**0.21854**
Proposed method (4)	0.22069	0.19512	0.26178	0.19497	0.24444	0.21831	0.21142	0.21881

Table 5 Details of estimation error averages (Domain: Terra)[†††].

Weighting Function	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
Unweighted	0.25053	0.31051	0.11317	0.15392	0.33007	0.27070	0.24404	0.33267
Upward-sloping linear function	0.25685	0.29891	0.11477	0.13454	0.37726	0.26107	0.26009	0.35162
Downward-sloping linear function	0.25086	0.31999	0.11516	0.16729	0.31359	0.27305	0.24198	0.32645
Downward-convex quadratic function	0.24927	0.30949	0.11665	0.17563	0.30066	0.27909	0.24364	0.32172
Upward-convex quadratic function	0.25697	0.31155	0.11357	0.14357	0.36164	0.26254	0.25573	0.35053
Exponential function	0.25475	0.29625	0.11652	0.13542	0.37206	0.26471	0.25375	0.34517
Proposed method (1)	0.17291	0.19615	**0.08078**	0.08287	0.22702	0.21782	0.18107	0.22767
Proposed method (2)	0.17560	0.21036	0.08089	0.08272	0.22083	0.22915	0.18019	0.22865
Proposed method (3)	**0.14639**	**0.15846**	0.08364	**0.07676**	0.18799	**0.20257**	**0.13660**	**0.18245**
Proposed method (4)	0.15165	0.17988	0.08370	0.07689	**0.18317**	0.21975	0.13708	0.18566

As shown in Table 7, the estimation errors of our proposed method weren't affected by the discount factor because the estimation accuracy of the opponents' utility functions in each learner is hardly affected by the discount factor. Usually, simple frequency-based methods are affected by discount factors because most agents try to compromise at an early stage when the discount factor is small. However, the frequency-based methods used as the learners in our proposed method consider the timeline changes.

In addition, our proposed method improved the accuracies of the estimation method of ParsCat, Farma, and Ngent more than the Unweighted method. The main reasons are that these agents forge compromises based on the discount factors. In other words, our method improves the accuracies of the opponent estimations because their agents proposed various bids in short terms.

[†††]Bold indicates the best performance per agent.

Table 6 Details of estimation error averages (Domain: Party_domain, Unweighted).

DF	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
0.1	0.28146	0.33485	0.13924	0.20370	0.36224	0.34198	0.22643	0.36240
0.2	0.28183	0.33405	0.13989	0.20156	0.36224	0.34661	0.22332	0.36512
0.3	0.27787	0.33307	0.13835	0.19570	0.35077	0.34756	0.21373	0.36593
0.4	0.27644	0.33393	0.13766	0.18731	0.35306	0.34875	0.20971	0.36469
0.5	0.27219	0.33329	0.13737	0.18055	0.34241	0.34857	0.20330	0.35984
0.6	0.27352	0.33109	0.13690	0.17929	0.34425	0.34864	0.20486	0.36959
0.7	0.27162	0.33049	0.13637	0.17165	0.33830	0.34871	0.20140	0.37444
0.8	0.26866	0.32547	0.13769	0.16031	0.33156	0.34849	0.20007	0.37700
0.9	0.26759	0.31781	0.13790	0.15292	0.33763	0.34802	0.19992	0.37895
1	0.26584	0.32705	0.13698	0.14979	0.29852	0.34136	0.20183	0.40534

Table 7 Details of estimation error averages (Domain: Party_domain, Proposed method (3)).

DF	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
0.1	0.17797	0.18979	0.10868	0.11646	0.22848	0.23592	0.13135	0.23495
0.2	0.17829	0.19049	0.10741	0.11426	0.23419	0.24245	0.12928	0.22996
0.3	0.17611	0.18929	0.10730	0.10981	0.22449	0.25175	0.12677	0.22337
0.4	0.17558	0.19123	0.10720	0.10626	0.22642	0.25466	0.12780	0.21551
0.5	0.17361	0.19058	0.10645	0.10325	0.21842	0.25757	0.12718	0.21182
0.6	0.17533	0.19388	0.10577	0.10520	0.21306	0.25830	0.12722	0.22390
0.7	0.17458	0.19589	0.10589	0.10453	0.20569	0.25687	0.12727	0.22589
0.8	0.17554	0.20611	0.10467	0.10347	0.20665	0.25796	0.12641	0.22347
0.9	0.17980	0.21641	0.10149	0.10035	0.24614	0.23866	0.13003	0.22549
1	0.17183	0.23165	0.10217	0.10175	0.18746	0.21213	0.12956	0.23808

On the other hand, the trends of our proposed method resemble the unweighting method by changing the discount factors and the opponents because our proposed method uses a frequency-based method.

Therefore, our proposed method can estimate opponents' utility functions by adjusting the combinations of each frequency-based method when a discount factor is changed.

5.5 Estimation Errors when Cooperativeness of Domains is Changed

5.5.1 Experimental Settings

ANAC2015 introduced a qualitative index that is referred to as Cooperativeness [27]. A cooperative domain includes bids through which all agents can obtain high utilities, and a competitive domain includes bids where the variances of the utilities of each agent are high even though the social welfare of the bids is high. We experimentally showed the performance of our method under different kinds of cooperative or competitive domains. Experiments were conducted to see the effect of estimation errors in Cooperativeness with agents used in ANAC2016 and some domains used in ANAC2015 (Table 8). Experimental settings are the same as in Sect. 5.2, except as follows:

- Domains: Domain ID 5, 6, 7, and 8 (used in ANAC2015).
- Negotiation time: 60 seconds.
- Total number of negotiations: $210 \times 4 = 840$.
- The proposed method (3) was used for our estimation method.

Table 8 Ten domains used for ANAC2015.

Domain ID	**Issues (size)**	**DF**	**RV**	**Cooperativeness**
1	1 (5 outcomes)	1.0	0.5	very competitive
2	1 (5 outcomes)	1.0	0.5	a bit cooperative
3	2 (25 outcomes)	0.2	0.0	very competitive
4	2 (25 outcomes)	1.0	0.5	quite collaborative
5	4 (320 outcomes)	0.5	0.0	competitive
6	4 (320 outcomes)	0.5	0.0	collaborative
7	8 (3^8 outcomes)	1.0	0.0	competitive
8	8 (3^8 outcomes)	1.0	0.0	collaborative
9	16 (2^{16} outcomes)	0.4	0.7	very collaborative
10	16 (2^{16} outcomes)	0.4	0.7	very competitive

5.5.2 Experimental Results

Tables 9 and 10 show the averages of the estimation errors. Our proposed method outperformed the simple unweighting method in all domains. In addition, its estimation errors are almost the same, even though the estimation errors of the existing method are affected by the cooperativeness of domains. This is because our proposed method adjusted the combination of some frequency-based methods when the domain's cooperativeness was changed.

Focusing on the collaborative domains, our proposed method drastically improved the estimation accuracies. This is because the opponents proposed more various bids (which can obtain high social welfare) in the collaborative domains than the competitive domains. On the other hand, the trends of our proposed method resemble the unweighting method for changing the cooperativeness of the domains because it used a frequency-based method. Therefore, our proposed method estimated opponent utility functions by adjusting the combinations of each frequency-based method when the

Table 9 Details of estimation error averages (Method: Unweighted).

ID	Cooperativeness	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
5	competitive	0.24145	0.22824	0.25844	0.14454	0.27020	0.22424	0.24216	0.32376
6	collaborative	0.27622	0.31804	0.18613	0.17485	0.36729	0.24824	0.28730	0.35170
7	competitive	0.22661	0.28475	0.09600	0.11027	0.26275	0.27343	0.16702	0.39206
8	collaborative	0.25215	0.31603	0.10452	0.14944	0.30765	0.27624	0.23780	0.37337

Table 10 Details of estimation error averages (Method: proposed method (3)).

ID	Cooperativeness	Average	Caduceus	YXAgent	ParsCat	Farma	MyAgent	Atlas3	Ngent
5	competitive	0.16337	0.15280	0.25770	0.10166	0.18201	0.11042	0.15981	0.17927
6	collaborative	0.16681	0.16676	0.16184	0.07763	0.21704	0.19065	0.15749	0.19624
7	competitive	0.15426	0.23216	0.10197	0.06330	0.16915	0.17245	0.11641	0.22439
8	collaborative	0.16804	0.24164	0.04060	0.05402	0.26701	0.20139	0.16235	0.20930

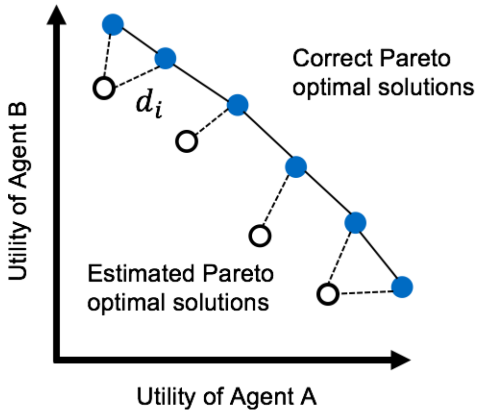

Correct Pareto optimal solutions
d_i
Estimated Pareto optimal solutions
Utility of Agent B
Utility of Agent A

Fig. 5 Inverted generational distance.

cooperativeness of the domains was changed.

5.6 Accuracies of Estimated Pareto Fronts

5.6.1 Experimental Settings

We conducted experiments to show the accuracy of our estimated Pareto fronts using our proposed method. First, we define some metrics for their accuracy.

(1) **Inverted Generational Distance (IGD)**
IGD is a metrics used in multi-objective optimization that indicates how much the estimated Pareto front resembles the correct Pareto front [28]. Figure 5 shows the concept of IGD, which can be calculated by formula (5):

$$IGD = \frac{1}{N}\sum_{i=1}^{N} d_i. \tag{5}$$

N means the number of Pareto optimal solutions that are included in the correct Pareto front. d_i is the Euclidean distance in the correct utility space between Pareto optimal solution i, which is included in the correct Pareto front, and the Pareto optimal solution nearest the estimated Pareto front.

(2) **F-score**
F-score is a measure of the estimation accuracy used in the statistical analysis, shown as formula (6):

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{6}$$

where *precision* is the estimated Pareto optimal solutions included in the correct Pareto fronts divided by the estimated Pareto optimal solutions. *recall* is the estimated Pareto optimal solutions included in the correct Pareto fronts divided by the correct Pareto optimal solutions.

We experimentally evaluated the accuracies of our method using these metrics with agents from ANAC2016 and domains from ANAC2015 (Table 8). Experimental settings are the same as in Sect. 5.2, except as follows:

- Domains: Domain ID 1 to 10 (used in ANAC2015).
- Negotiation time: 60 seconds.
- Total number of negotiations: $210 \times 10 = 2100$.
- Proposed method (3) was used for this experiment.

5.6.2 Experimental Results

Table 11 shows the details of the average metrics related to the estimated Pareto fronts. The average metrics of the proposed method improved in many cases. In particular, our method significantly improved the F-score in Domains 1 and 2 because it can estimate the bids with smaller utilities than the existing method.

In some cases, the IGD score is worse than the existing method, because our method searched for a non-optimal solution as a Pareto optimal solution.

In addition, our method couldn't drastically outperform the existing method in domains that are competitive and/or large number of issues. The main reason is that our estimation method can't receive many kinds of bids in the negotiation. Our proposed method needs to receive various domains to estimate Pareto fronts based on domain sizes and bid distributions.

Table 11 Details of average metrics related to estimated Pareto fronts[††††].

DomainID	Competitiveness	Method	Precision	Recall	F-score	IGD
1	Very competitive	Unweighted	**0.9918**	0.5523	0.6841	0.4315
1	Very competitive	Proposed method (3)	0.9898	**0.8933**	**0.9294**	**0.1695**
2	Slightly cooperative	Unweighted	0.9612	0.5112	0.6467	0.3736
2	Slightly cooperative	Proposed method (3)	**0.9945**	**0.8755**	**0.9196**	**0.2043**
3	Very competitive	Unweighted	0.8856	0.6905	0.7503	0.1358
3	Very competitive	Proposed method (3)	**0.9000**	**0.7159**	**0.7733**	**0.0885**
4	Quite collaborative	Unweighted	0.7856	0.6248	0.6665	0.1216
4	Quite collaborative	Proposed method (3)	**0.7980**	**0.6677**	**0.7069**	**0.0837**
5	Competitive	Unweighted	0.4910	0.5179	0.4714	**0.0895**
5	Competitive	Proposed method (3)	**0.4960**	**0.5342**	**0.4857**	0.1165
6	Collaborative	Unweighted	0.6111	0.5928	0.5813	0.0788
6	Collaborative	Proposed method (3)	**0.6207**	**0.6038**	**0.5925**	**0.0654**
7	Competitive	Unweighted	**0.4505**	**0.3675**	**0.3872**	0.0476
7	Competitive	Proposed method (3)	0.4472	0.3655	0.3854	**0.0461**
8	Collaborative	Unweighted	0.5303	0.5644	0.5214	0.0683
8	Collaborative	Proposed method (3)	**0.5332**	**0.5705**	**0.5270**	**0.0462**
9	Very collaborative	Unweighted	**0.5645**	0.2488	0.3171	0.0445
9	Very collaborative	Proposed method (3)	0.5504	**0.2612**	**0.3280**	**0.0372**
10	Very competitive	Unweighted	**0.4431**	0.2986	0.3123	**0.0512**
10	Very competitive	Proposed method (3)	0.4245	**0.3127**	**0.3171**	0.0573

5.7 Summary of Experimental Results

The following summarizes our experimental results:

- Our proposed method outperformed the existing frequency-based methods.
- It estimated the opponent utility functions in domains with various discount factors and cooperativeness.
- It accurately estimated the Pareto front compared with the existing method in various domains.

6. Conclusion

This paper focused on multi-time negotiations that are conducted several times under the conditions of the same domains and opponents. We proposed a method of estimating the opponents' utility functions by combining several existing methods of estimating the opponents' utility functions using boosting based on the least-squares method and nonlinear programming. Furthermore, we proposed a method to estimate a Pareto front using the estimated opponents' utility function. The experimental results demonstrated that the estimation accuracy of the opponents' utility functions was improved in many cases by combining several existing estimation methods. In addition, the average estimation errors were reduced by about 69% in the best cases compared with the existing methods.

One of the possible future works is the improvement of the proposed estimation method of the opponents' utility functions. Such improvements are anticipated by adopting other effective utility function estimation methods. Another possible future work is to propose the negotiation strategy based on our proposed estimation method using boosting. For this purpose, the estimation of an opponent's compromising and the utility function are utilized at the same time, and self-evaluation of the estimation should be effective.

[††††]Bold indicates the best performance per domain.

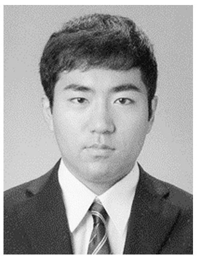
Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR15E1, Japan.

References

[1] T. Matsune and K. Fujita, "Weighting estimation methods for opponents' utility functions using boosting in multi-time negotiations," Proc. 2nd International Conference on Agents (ICA 2017), pp.27–32, 2017.
[2] T. Baarslag, "Accepting optimally with incomplete information," Exploring the Strategy Space of Negotiating Agents: A Framework for Bidding, Learning and Accepting in Automated Negotiation, Springer Theses, pp.91–109, Springer International Publishing, Cham, 2016.
[3] S. Kraus, Strategic Negotiation in Multiagent Environments, MIT Press, 2001.
[4] S. Kraus, J. Wilkenfeld, and G. Zlotkin, "Multiagent negotiation under time constraints," Artificial Intelligence, vol.75, no.2, pp.297–345, 1995.
[5] X. Luo, C. Miao, N.R. Jennings, M. He, Z. Shen, and M. Zhang, "KEMNAD: A knowledge engineering methodology for negotiating agent development," Computational Intelligence, vol.28, no.1, pp.51–105, 2012.
[6] R. Aydogan, K. Fujita, T. Baarslag, K. Hindriks, T. Ito, and C. Jonker, The Seventh International Automated Negotiating Agents Competition, 2016.
[7] P. Faratin, C. Sierra, and N.R. Jennings, "Using similarity criteria to make issue trade-offs in automated negotiations," Artificial Intelligence, vol.142, no.2, pp.205–237, 2002.
[8] S. Zhou, S. Chen, E. Smirnov, G. Weiss, and K. Tuyls, "Automatic knowledge transfer for agent-based bilateral negotiation tasks," International Workshop on Agent-Based Complex Automated Negotiations (ACAN2014), 2014.
[9] S. Chen, S. Zhou, G. Weiss, and K. Tuyls, "Using transfer learning to

model unknown opponents in automated negotiations," Recent Advances in Agent-Based Complex Automated Negotiation, Studies in Computational Intelligence, vol.638, pp.175–192, Springer International Publishing, Cham, 2016.
[10] R.E. Schapire, "The strength of weak learnability," Machine Learning, vol.5, no.2, pp.197–227, 1990.
[11] M. Ikarashi and K. Fujita, "Compromising strategy using weighted counting in multi-times negotiations," IIAI 3rd International Conference on Advanced Applied Informatics (IIAI AAI2014), pp.453–458, 2014.
[12] T. Baarslag, K. Hindriks, C. Jonker, S. Kraus, and R. Lin, "The first automated negotiating agents competition (ANAC2010)," New Trends in Agent-Based Complex Automated Negotiations, Studies in Computational Intelligence, vol.383, pp.113–135, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[13] R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, and C.M. Jonker, "Genius: An integrated environment for supporting the design of generic automated negotiators," Computational Intelligence, vol.30, no.1, pp.48–70, 2014.
[14] T. Baarslag, K. Fujita, E.H. Gerding, K. Hindriks, T. Ito, N.R. Jennings, C. Jonker, S. Kraus, R. Lin, V. Robu, and C. Williams, "Evaluating practical negotiating agents: Results and analysis of the 2011 international competition," Artificial Intelligence Journal, vol.198, pp.73–103, 2013.
[15] S. Chen and G. Weiss, "An efficient and adaptive approach to negotiation in complex environments," Proc. 19th European Conference on Artificial Intelligence (ECAI2012), pp.228–233, 2012.
[16] S. Chen, H.B. Ammar, K. Tuyls, and G. Weiss, "Conditional restricted Boltzmann machines for negotiations in highly competitive and complex domains," Proc. 23th International Joint Conference on Artificial Intelligence (IJCAI2013), pp.69–75, 2013.
[17] C.R. Williams, V. Robu, E.H. Gerding, and N.R. Jennings, "Using Gaussian processes to optimise concession in complex negotiations against unknown opponents," Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI2011), pp.432–438, 2011.
[18] T. Baarslag and K. Hindriks, "Accepting optimally in automated negotiation with incomplete information," Proc. 2013 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2013), pp.715–722, 2013.
[19] R. Aydoğan, D. Festen, K.V. Hindriks, and C.M. Jonker, "Alternating offers protocols for multilateral negotiation," Modern Approaches to Agent-Based Complex Automated Negotiation, Studies in Computational Intelligence, vol.674, pp.153–167, Springer International Publishing, Cham, 2017.
[20] A. Rubinstein, "Perfect equilibrium in a bargaining model," Econometrica: Journal of the Econometric Society, vol.50, no.1, pp.97–109, 1982.
[21] H. Shinohara and K. Fujita, "Compromising strategy using analytic hierarchy process for multiparty closed automated negotiations," International Workshop on Agent-Based Complex Automated Negotiations (ACAN2016), pp.44–51, 2016.
[22] Z. Khosravimehr and F. Nassiri-Mofakham, "Pars agent: Hybrid time-dependent, random and frequency-based bidding and acceptance strategies in multilateral negotiations," Modern Approaches to Agent-Based Complex Automated Negotiation, Studies in Computational Intelligence, vol.674, pp.175–183, Springer International Publishing, Cham, 2017.
[23] S. Kakimoto and K. Fujita, "Preliminary estimating method of opponent's preferences using simple weighted functions for multilateral closed multi-issue negotiations," Multi-agent and Complex Systems, ed. Q. Bai, F. Ren, K. Fujita, M. Zhang, and T. Ito, Studies in Computational Intelligence, vol.670, pp.181–192, Springer Singapore, Singapore, 2017.
[24] S. Kakimoto and K. Fujita, "RandomDance: Compromising strategy considering interdependencies of issues with randomness," Modern Approaches to Agent-Based Complex Automated Negotiation, Studies in Computational Intelligence, vol.674, pp.185–189, Springer International Publishing, Cham, 2017.
[25] H. Shinohara and K. Fujita, "Alternating offers protocol considering fair privacy for multilateral closed negotiation," PRIMA 2017: Principles and Practice of Multi-Agent Systems, ed. B. An, A. Bazzan, J. Leite, S. Villata, and L. van der Torre, Lecture Notes in Computer Science, vol.10621, pp.533–541, Springer International Publishing, Cham, 2017.
[26] S.S. Fatima, M.J. Wooldridge, and N.R. Jennings, "Multi-issue negotiation with deadlines," J. Artif. Int. Res., vol.27, pp.381–417, Nov. 2006.
[27] R. Aydogan, T. Baarslag, K. Fujita, K. Hindriks, T. Ito, and C. Jonker, "The sixth international automated negotiating agents competition," Modern Approaches to Agent-Based Complex Automated Negotiation, Studies in Computational Intelligence, vol.674, pp.139–151, 2015.
[28] P. Czyżak and A. Jaszkiewicz, "Pareto simulated annealing—A metaheuristic technique for multiple-objective combinatorial optimization," Journal of Multi-Criteria Decision Analysis, vol.7, no.1, pp.34–47, 1998.



Takaki Matsune is a master student of faculty of Engineering, Tokyo University of Agriculture and Technology. He received the B.S. degree from Departments of Computer and Information Sciences, Faculty of Engineering, Tokyo University of Agriculture and Technology, Japan, in 2017. His research interests are in automated negotiation and multi-agent systems.

Katsuhide Fujita is an Associate Professor of Faculty of Engineering, Tokyo University of Agriculture and Technology. He received the B.E., M.E, and Doctor of Engineering from the 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-agent systems and decision support systems.