Journal of Information Processing Vol.32 2–9 (Jan. 2024)

[DOI: 10.2197/ipsjjip.32.2]

Regular Paper

Effective Acceptance Strategy Using Deep Reinforcement Learning in Bilateral Multi-issue Negotiation

Hyuga Matsuo[1,a) Katsuhide Fujita[2,b)

Received: March 14, 2023, Accepted: October 3, 2023

Abstract: Recently, automated negotiation has been attracting attention in multi-agent systems to resolve conflicts and reach an agreement among agents. In automated negotiation, two main types of strategies are incorporated in each agent: a bidding strategy that considers what kind of bid to send to an opponent, and an acceptance strategy that considers whether to accept the opponent's offer. In most bilateral multi-issue negotiation, agents take turns sending bids to each other and the negotiation ends when an agent accepts an opponent's offer. Therefore, the acceptance strategy is important in terms of increasing the utility of an agent. However, most studies of automated negotiation using reinforcement learning focus only on the bidding strategy of the agent, so there are not many studies that investigate acceptance strategies using reinforcement learning. In this paper, we propose a new configuration of a deep reinforcement learning framework for the acceptance strategy in automated negotiations using Deep Q-Network. The training phase is performed multiple times with various reward functions, and the reward capable of a higher utility value is investigated. Simulation experiments with other negotiating agents showed that the proposed method obtained significantly higher utility values than existing methods.

Keywords: acceptance strategy, automated negotiation, multi-agent systems, deep reinforcement learning

1. Introduction

Recently, multi-issue negotiation has piqued interest in multi-agent system research in reaching an agreement among multiple agents. Each agent behaves autonomously based on its own preferences and conflicts therefore occur among agents. In such cases, automated negotiation has been attracting attention as a means of resolving conflicts and reaching an agreement among agents. Automated negotiation technologies can be used in supply chains [1], smart grids [2], and automated driving [3].

The International Automated Negotiating Agents Competition (ANAC) has been held yearly since 2010 to promote automated negotiation research through competitions [4]. In ANAC, negotiations are conducted under various conditions based on negotiation strategies created by each participant, and each agent competes for its individual utility and social welfare to win the competition. In addition, various negotiation strategies are proposed in ANAC (Refs. [5], [6]), and the sharing of submitted strategies of agents contributes to the development and evaluation of new negotiation strategies. Genius [7], which was used as a negotiation platform in the Agent-Agent negotiations, provides a common platform for automated negotiation including standardized negotiation protocols and scenarios.

In automated negotiation, two main types of strategies are incorporated into an agent: a bidding strategy that considers what kind of bid to send to an opponent, and an acceptance strategy that considers whether to accept the opponent's bid. In bilateral multi-issue negotiation, each agent takes turns sending an offer to another agent. If either agent accepts the other's offer, the negotiation ends and the utility value of the offer in each agent is obtained. Therefore, an acceptance strategy determining when and what kind of proposal from an opponent to accept is important in terms of increasing its utility.

In recent automated negotiation research, some existing studies using reinforcement learning for agent bidding strategy have been investigated (Refs. [8], [9], [10], [11] etc.). On the other hand, most of the acceptance strategies in existing works are based on the heuristics approach [12] that do not use machine learning. Heuristic acceptant strategies typically decide whether to accept or reject a bid (offer) based on a predefined equation with heuristic parameters considering the negotiation time or utility. However, heuristic strategies might not be capable of obtaining high utility depending on the bidding strategy of the opponent. Recently, a deep reinforcement learning approach for obtaining acceptance strategies has been proposed [13]. The proposed method demonstrated the effectiveness of the proposed method by setting the inputs and reward functions effectively in the training phase. In the existing work, the reinforcement learning algorithm that learns whether or not to accept a bid sent by the opponent is a Deep Q-Network (DQN). However, in the experiments, other methods were able to outperform reinforcement learning. This situation makes it necessary to reconsider the input and reward function in order to improve the utility compared to method without learning functions.

[1] Tokyo University of Agriculture and Technology, Koganei, Tokyo 184–8588, Japan

[2] The author is with Institute of Global Innovation Research, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184–8588, Japan

a) matsuo@katfuji.lab.tuat.ac.jp

b) katfuji@cc.tuat.ac.jp

The main contributions of this study are summarized below.

- We propose a deep reinforcement learning framework for an acceptance strategy in bilateral multi-issue negotiations.
- We demonstrate that our proposed approach can obtain higher utility values than the existing methods through simulation experiments.
- We investigate the rewards of the reinforcement learning framework for acceptance strategies by comparing the obtained utility values in some experiments. We compare three types of rewards for learning: accepting the bid, continuing the negotiation without accepting the bid, and not completing the negotiation during the limited round.
- We propose and evaluate several patterns of reward functions for each type. Each combination of the proposed rewards is learned and evaluated in a simulation experiment.

The main factors for determining the performance of the negotiation strategy are the negotiation domains and the opponents. In particular, the degree of opposition to the negotiation domain is directly related to the ease of reaching an agreement. Therefore, we prepare more various negotiation domains focusing on the degree of opposition in our experiments to avoid the limited experimental settings.

We evaluate the performances of the proposed negotiation strategies based on the negotiation simulations, not theoretical analysis. This is because theoretical analysis frameworks for a bilateral multi-issue negotiation haven't been decided and the negotiation domains and the opponents influence its performance directly. Competitions (ANAC) also evaluate negotiation strategies, not theoretically in automated negotiation research. Thus, we decided that the theoretical analysis of the proposed approach is out of the scope of this paper.

2. Related Work

The International Automated Negotiating Agents Competition (ANAC) is a competition, where participants negotiate under various conditions using their automated negotiation agents and compete on their individual utility and social welfare (Refs. [4], [14]). ANAC has been held yearly since 2010. The main purpose of this competition to provide an incentive for the development of effective and efficient negotiation protocols and strategies for bidding, accepting and opponent modeling for different negotiation scenarios. This competition also develops a benchmark of negotiation scenarios, protocols and strategies to provide a common set of tools and criteria for the evaluation.

The BOA framework [15] is a decoupled negotiation strategy framework that divides the strategies of automated negotiation agents into three components: Bidding Strategy, Opponent Model, and Acceptance Strategy, each of which can be developed separately.

Bidding Strategy is a strategy for determining the next bid to be proposed by the agent. The agent determines the target utility value using the proposed strategy and proposes a bid with a utility sufficiently close to the target utility value. Typical bidding strategies include the following:

- Time-Dependent: A strategy in which the target utility value is determined depending on the negotiation time.
- Behavior Dependent: A strategy in which the target utility value is determined by the behavior of the negotiating agent.

Opponent Model is a strategy that estimates the utility function of the opponent using bids proposed by the opponent agent. Estimating the utility function of the opponent increases the likelihood that the opponent will accept your bid, that the social welfare will be greater, and so on.

Acceptance Strategy is a strategy for deciding whether to agree and terminate negotiations or reject and continue negotiations in response to a bid from the opponent agent. Typical acceptance strategies include the following [12]:

- Utility based: A strategy that is determined based on the utility value of the proposed bid.
- Time based: A strategy in which decisions are made based on negotiation time.

This paper focuses on the acceptance strategy in the BOA framework.

Bidding strategy using reinforcement learning

Sunder et al. [8] used reinforcement learning to decide what bid it will send. Their methods learn the content of the bid and not the utility value.

Bakker et al. [9] proposed the RLBOA framework, which allows learning to be performed with the strategies of agents separated. They also learned the bidding strategies using Q-learning in this framework. Their method defines 10 utility intervals, and the agent learns which utility value belongs to which interval to use for the next bid.

Ayan et al. [16] proposed a method for sensing changes in the utility function and creating bidding strategies that adapt to them. They used bidirectional Long Short-Term Memory [17] for learning and transfer learning for creating adapted strategies. They switched the proposed strategy to obtain a higher utility value.

These methods focus on reinforcement learning frameworks for bidding strategies, not acceptance strategies. On the other hand, this paper focuses on learning different acceptance strategies.

Acceptance strategy

Baarslag et al. [12] proposed an effective acceptance strategy by combining predefined rules. For example, "AC–next," in which an agent accepts an opponent's offer if the utility of the offer is higher or equal to the utility of his next offer, has been used in practice in many studies. Their method can be expected to provide good utility values based on the heuristics but it does not utilized machine learning.

3. Baseline Algorithm

Razeghi et al. [13] proposed an acceptance strategy in automated negotiation using deep reinforcement learning and it is the only existing work focusing on the acceptance strategy with reinforcement learning. Their method uses a DQN, and the output is whether to accept or reject the opponent's bid. In addition, the following five elements are defined as inputs.

- ΔO: The difference between the utility value obtained from the opponent's bid and the reservation value.
- D: Normalized negotiation time.
- MNU: The utility value that I plan to propose next.

- R: The target utility value.
- C: Current utility value of the opponent's bid.

The target utility value R is determined heuristically, which is 0.8 in their approach.

The reward function is calculated as the immediate reward defined by Eq. (1):

$$r = \begin{cases} -2^{|0.8-f|} & (0.8 > f) \\ 2^{|0.8-f|} & (0.8 \leq f) \\ 0 & \textit{when negotiation continues} \end{cases} \tag{1}$$

where f represents the utility value obtained when the agreement is made, and *when negotiation continues* is when the opponent rejects the bid. Their method has the limitation of obtaining only the same level of utility as existing methods that do not use learning.

4. Bilateral Multi-issue Negotiation Problem

4.1 Negotiation Domain

In the bilateral multi-issue negotiation problem, two agents negotiate on a common domain. A negotiation domain is defined by the issues in the negotiations and the options for each issue and consists of n issues $I_1, I_2, \ldots, I_n$ and k_i options $v_1^i, v_2^i, \ldots, v_{k_i}^i$ in each issue I_i. A bid, which is offered during negotiations, is a selection of one option from the value of each issue, denoted by $\omega = \left[v_{x_1}^1, v_{x_2}^2, \ldots, v_{x_n}^n\right]$. Agents offer bids to each other according to a bidding strategy to reach a single agreement.

An example of a negotiation domain is shown in **Table 1**. The example domain has two issues, "travel destination" and "travel expenses," with three options for the issue "travel destination," including "Sapporo," and two options for the issue "travel expenses." Examples of bids in this domain include [Sapporo, 200,000 yen] and [Okinawa, 50,000 yen].

4.2 Utility Function

Each negotiating agent has its utility function. A function is a weighted-sum function that outputs a utility value for each bid based on the weights of the issues and the evaluated value of the options on each issue in this paper. The utility function of each agent is private and not known to other agents.

The utility function $U(\omega)$ for a given bid ω is expressed by Eq. (2).

$$U(\omega) = \sum_{i=1}^{n} w_i \times \frac{eval(v_{x_i}^i)}{\max_j eval(v_j^i)} \tag{2}$$

where w_i denotes the weight of issue I_i, and $eval(v_x^i)$ denotes the evaluation value for option v_x^i. However, the weight of the issue w_i should satisfy $\sum_{i=1}^{n} w_i = 1$ and $w_i \geq 0$, and the evaluation value should satisfy $eval(v_x^i) \geq 0$. The value obtained by the utility function $U(\omega)$ is called the utility value of bid ω and is expressed as a real number between 0 and 1. The objective function of the bilateral multi-issue negotiation in this paper is to maximize its utility value.

Table 1 Example of negotiation domain.

Issues	Options
Travel destination	Sapporo Nagoya Okinawa
Travel expenses	200,000 yen 50,000 yen

The reservation value is the minimum utility value that can be obtained when the negotiation is terminated without reaching a negotiated agreement.

4.3 Alternating Offers Protocol

Here we consider Alternating Offers Protocol (AOP) [18], which is widely used in bilateral multi-issue negotiations. In AOP, two agents take turns performing actions. In its turn, the agent selects the action from the following three actions.

- Accept: Accept the last bid proposed by the opponent.
- Offer: Reject the opponent's bid and propose a new bid to the opponent.
- EndNegotiation: Terminate negotiations without agreement.

The negotiation deadline is defined as the actual execution time or the number of rounds (the number of bid proposals of each agent). The agents repeat their actions until the deadline is reached, and the negotiation ends when one of the agents accepts the offer, one of the agents selects EndNegotiation, or the deadline is reached. After the negotiation, the agents receive the utility value of the bid if an agreement has been reached or the reservation value if no agreement has been reached.

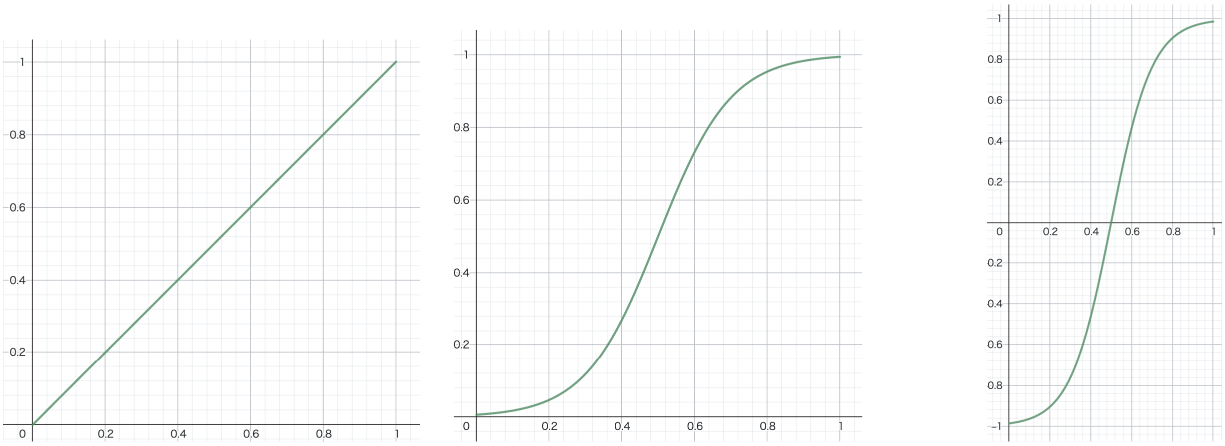
5. Deep Reinforcement Learning Framework for Acceptance Strategy

We propose the deep reinforcement learning framework for the acceptance strategy, composed of the following elements: state, action, and reward. The pseudo code for DQN is shown in Algorithm 1. In our proposed method, the actions are either "Accept" or "Reject," and the bidding strategy is that of AgentK [5]. These two are the same as in the existing work [13]. In the proposed method, inputs and rewards are changed following the existing work [13].

Algorithm 1 Acceptance strategy of based on Q-learning

```
$S_{now} \leftarrow getCurrentState()$
$< V_A, V_R > \leftarrow NN.predict(S_{now})$
if $1 - \epsilon$ probablity then
    if $V_A > V_R$ then
        $Act \leftarrow Accept$
    else
        $Act \leftarrow Reject$
    end if
else
    $Act \leftarrow Random(Accept|Reject)$
end if
$r \leftarrow RewardFunction(Act)$
$S_{next} \leftarrow getNextState()$
$NN.train(S_{next}, S_{now}, r, V_A, V_R)$
return $Act$
```

- V_A: The Q-value of accept action on current state.
- V_R: The Q-value of reject action on current state.
- NN: The Neural Network.



(a) Equation (4-a)

(b) Equation (4-b)

(c) Equation (4-c)

Fig. 1 Rewards vary with the utility value of each equation.

5.1 Input

We reduce the number of input elements in the existing work and use an input consisting of the following information:

- Utility value of the opponent's bid.
- Normalized negotiation time.

The state at time t is expressed using Eq. (3).

$$\left[U_A\left(\omega_{B\to A}^{t-1}\right), \frac{t}{T}\right] \tag{3}$$

When focusing only on the acceptance strategy, the utility value of the opponent's bid and the negotiation time are considered important for increasing one's utility value. Existing work [12] has shown that an acceptance strategy considering the two factors (the utility value of the opponent's bid and the negotiation time) had good performances. Therefore, we define the improved input that contains only two factors to reduce the size of the input spaces in the reinforcement learning framework.

5.2 Reward Function

The reward function is divided into the following three parts, each of which is considered independently.

- Reward at the time of reaching an agreement.
- Reward at the time of continued negotiations.
- Penalty for failure to negotiate.

We propose the following three types of rewards at the time of reaching an agreement, as shown in Eqs. (4-a)–(4-c).

$$r_a^{accept} = U_A\left(\omega_{B\to A}^{t-1}\right) \tag{4-a}$$

$$r_b^{accept} = \frac{\tanh\left\{5\left(U_A\left(\omega_{B\to A}^{t-1}\right) - 0.5\right)\right\} + 1}{2} \tag{4-b}$$

$$r_c^{accept} = \tanh\left\{5\left(U_A\left(\omega_{B\to A}^{t-1}\right) - 0.5\right)\right\} \tag{4-c}$$

In this case, Eq. (4-a) uses the obtained utility value as the reward, and Eqs. (4-b) and (4-c) calculate the values from the obtained utility value. Equation (4-b) is a function based on Eq. (4-a), where the reward is lower when the obtained utility value is less than 0.5, and higher when it is greater than 0.5. Furthermore, Eq. (4-c) is a function that extends Eq. (4-b) in the negative direction. For example, the obtained utility value is -1 when the value is 0. The obtained utility value is 0 when the value is 0.5. With these types of transformations, we expect to obtain higher utility values than Eq. (4-a). **Figure 1** shows how the rewards vary with the utility value of each equation.

The reward at the time of continued negotiations is calculated in rejecting the opponent's bid and proposing a new bid to the opponent, i.e., for making an offer. We propose three types of reward functions in continuing the negotiation shown in Eqs. (5-a)–(5-c).

$$r_a^{offer} = 0 \tag{5-a}$$

$$r_b^{offer} = \frac{1 - U_A\left(\omega_{B\to A}^{t-1}\right)}{100} \tag{5-b}$$

$$r_c^{offer} = \frac{1 - average}{100} \tag{5-c}$$

Note that *average* in Eq. (5-c) is the average of the last 10 utility values of the bid received from the opponent. Equation (5-a) gives no reward in continuing the negotiation, whereas Eqs. (5-b) and (5-c) give larger rewards for smaller utility values of the bids received or their averages. This mechanism is intended to avoid a situation where the utility of the received bid is small, that is: a situation in which only a small utility is obtained by accepting the bid now, and to obtain a higher utility by using the specification of reinforcement learning to maximize the accumulated reward. In addition, because the most important factor in this case is the high utility value at the time of agreement formation, the reward is multiplied by one hundredth to ensure that the reward is unaffected.

The penalty for negotiation failure is the negative reward earned if the negotiations reach the deadline without reaching an agreement. We propose three types of reward functions: -1, -0.5, and 0. We expect that these rewards will help us to learn to avoid negotiation failures.

6. Experimental Results

We demonstrate the performance of the proposed method compared with the existing work. We use obtained individual utility as the evaluation metric in our experiments.

The environment for the simulation is created in OpenAI Gym [19], and NegMAS [20] is used as the negotiation platform.

In addition, we propose an acceptance strategy only for reinforcement learning and we therefore use the existing bidding strategy.

In addition, the deadline is 100 rounds per negotiation throughout all training and experiments. In this setup, negotiations are terminated when both agents have sent 100 bids each, due to the deadline.

6.1 Experimental Settings

Training Setup

Training is performed independently for each domain, opponent agent, and reward function combination. In this experiment, there is one domain, two opponent agents, and 3^3 reward functions; thus, a total of $1 \times 2 \times 3^3 = 54$ patterns are learned. In addition, 10 independent training sessions are performed per pattern, including the baseline strategy. This training is intended to measure the stability of learning with multiple training sessions in the same settings and investigate the variation in the learning results.

Domains for Training: To compare the exact impact of the reward function, the domains in the training phase should be the same as those in the test phase in the experiments. Therefore, we use the England-Zimbabwe domain, which is the same as the domain used in the existing work [13]. The reservation value is 0 and the utility function does not consider discounting. When the reservation value is 0, the utility value obtained upon negotiation failure is 0.

Opponent Agents' Strategies for Training: Two types of opponent agents are used for learning: Gahboninho [21] and a time-dependent strategy. In this case, the opponent agent only does the bidding and does not accept the offered bids because its own agent learns to accept bids.

Gahboninho has a very selfish and stubborn strategy, making favorable offers until just before the deadline and later conceding. **Figure 2** shows the evolution of the utility value of the bid proposed by this agent. The agent is also used for training the existing work [13].

A time-dependent strategy that makes gradual concessions as the deadline is approached is used. In this strategy, the utility value $U_A\left(\omega^{t+1}_{A\to B}\right)$ of the next proposed bid $\omega^{t+1}_{A\to B}$ is determined using Eq. (6),

$$U_A\left(\omega^{t+1}_{A\to B}\right) = U_{max} - (U_{max} - U_{min}) \times \left(\frac{t}{T}\right)^{\frac{1}{e}} \qquad (6)$$

where U_{max} denotes the maximum value of the proposing utility, and U_{min} denotes the minimum value of the proposing utility. e denotes a parameter that determines the rate of concession and is a positive real number. For e, an agent that takes $e < 1$ is called a Boulware agent. In this experiment, $U_{max} = 1$, $U_{min} = 0$, and $e = 0.25$, and **Fig. 3** shows the trajectory of the utility values of the bid proposed by this agent.

Common Experimental Settings

Negotiations were performed 1,000 times per pattern under the same conditions for each experiment on agents trained using the training setup described previously. 100 negotiations for each of the 10 learned models per pattern. Here, we consider the average utility value obtained from these 1,000 negotiating sessions.

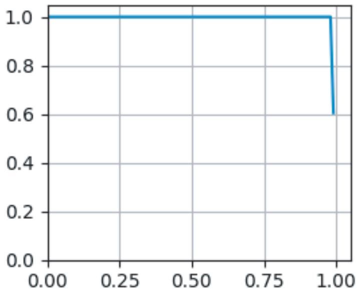


Fig. 2 Changes in Gahboninho's bidding utility values.

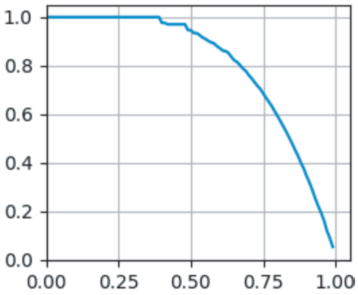


Fig. 3 Changes in Boulware's bidding utility values.

Table 2 Degrees of oppositions of domains in experiments.

Domain name	Degree of oppositions
Thompson	0.32557
England-Zimbabwe	0.27911
Party	0.26179
Kitchen	0.15243

The domain is selected on the basis of the measure of the degree of opposition. The degree of opposition is a measure of the ease of agreement: the closer it is to 0, the easier it is for both agents to obtain a high utility value, and the closer it is to 1, the easier it is for one side to obtain a good utility value. **Table 2** shows the degree of opposition in the domains used in the experiments. In addition, the opponent agent will not accept the offered bid because it accurately shows the learning results.

Baseline

As a baseline for evaluating the proposed method, we use the existing reinforcement learning approach for the acceptance strategy proposed by Razeghi et al. [13]. The input of the baseline is the five elements described in Section 3, and the state at time t is expressed using Eq. (7),

$$\left[U_A\left(\omega^{t-1}_{B\to A}\right) - RV, \frac{t}{T}, U_A\left(\omega^{t}_{A\to B}\right), 0.8, U_A\left(\omega^{t-1}_{B\to A}\right)\right] \qquad (7)$$

where $U_A(\omega)$ denotes the utility value of bid ω by its utility function, $\omega^{t-1}_{B\to A}$ denotes the bid that the opponent proposed immediately before, RV denotes the reservation value, T represents the deadline, and $\omega^{t}_{A\to B}$ represents the bid that he is going to propose next.

The reward function is defined by Eq. (1), and the possible actions are "Accept" or "Offer." The bidding strategy of the baseline is based on AgentK [5].

6.2 Experimental Results

6.2.1 Performances in Learning under Different Situations

In this experiment, we evaluate which reward pattern works for the other conditions by negotiating with the learned agent us-

Table 3 Average of utility values from negotiation under different conditions of learning. The bold means the best three average utility values of the agents learned by each opponent agent.

(a) Gahboninho, Penalty:−1

	(4-a)	(4-b)	(4-c)
(5-a)	0.547375	0.547424	0.639808
(5-b)	0.609366	0.639813	0.682915
(5-c)	0.616803	0.633991	0.663768

(b) Gahboninho, Penalty:−0.5

	(4-a)	(4-b)	(4-c)
(5-a)	0.542228	0.546344	0.652647
(5-b)	0.623362	0.631737	0.675719
(5-c)	0.624411	0.623424	**0.691335**

(c) Gahboninho, Penalty:0

	(4-a)	(4-b)	(4-c)
(5-a)	0.544045	0.550393	0.640888
(5-b)	0.633424	0.647326	**0.705536**
(5-c)	0.647955	0.663692	**0.712638**

(d) Boulware, Penalty:−1

	(4-a)	(4-b)	(4-c)
(5-a)	0.546290	0.566362	0.762705
(5-b)	0.751299	0.703250	**0.813732**
(5-c)	0.760013	0.717817	0.768888

(e) Boulware, Penalty:−0.5

	(4-a)	(4-b)	(4-c)
(5-a)	0.548513	0.556933	0.773661
(5-b)	0.774031	0.779451	0.745067
(5-c)	0.799192	0.743893	0.791371

(f) Boulware, Penalty:0

	(4-a)	(4-b)	(4-c)
(5-a)	0.549875	0.555964	0.701009
(5-b)	**0.824866**	0.754491	0.771464
(5-c)	**0.820254**	0.774558	0.722424

(g) Baseline

Existing method
0.699754

ing different domains and opponents not used for learning. The Party domain is used for negotiations. This domain is included in Genius and was used in the experiments because its degree of opposition and bid distribution are similar to those of the England-Zimbabwe domain used in the training. AgentK, the winner of ANAC, was used as the opponent agent [5].

Table 3 shows that, overall, the average of the obtained utility values is lower for those learned with Gahboninho than for those learned with Boulware. Compared with the baseline, only two patterns learned with Gahboninho are larger, whereas Boulware is larger than those for all patterns except when the reward is Eqs. (5-a) and (4-a) or (4-b). In particular, Welch's t-test at the 1% significance level confirmed that the top three patterns of the average utility values for those learned with Boulware were significantly different from the baseline. From the results in Table 3, the reward function including Eqs. (4-a), (4-c), (5-b), or (5-c) is effective for learning.

One reason why the average utility value varies depending on the agent used for learning is that a variety of inputs were provided during the training phase. Figure 2 and Fig. 3 show that Gahboninho proposes only approximately two types of bids, whereas Boulware proposes a variety of bids. This means that Gahboninho cannot learn from a variety of inputs. In such a case, when a bid with a certain utility value is made during the negotiation, it is likely to be accepted without waiting for another bid with a higher utility value. Therefore, its average is low. Conversely, when learning with a variety of inputs, the average is higher because the agent tends to wait for a bid with a higher utility value.

In addition, the reason why learning is not successful when the reward is Eqs. (5-a) and (4-a) or (4-b) is that no exploration of bids has occurred. In DQN, the learning is performed so that the final cumulative reward is maximized. Therefore, in cases learned with Eqs. (5-b) and (5-c), where the rewards for continuing to negotiate are positive, and the cases learned with Eq. (4-c), where the rewards for reaching an agreement include negative values, an attempt is made to continue negotiating longer to maximize the cumulative reward. In the process, it is possible to find a bid with a high utility value. However, when the reward for continuing the negotiation is zero and the reward for reaching an agreement is only positive, the agent can be satisfied with the current situation and will not continue for a long time. Therefore, it will not be able to find a bid with a high utility value.

The baseline method is lower than the proposed method for the above two reasons. The baseline method is trained using Gahboninho and does not have a variety of inputs. In addition, the reward for continuing the negotiation is zero, as shown in Eq. (1), and thus not enough exploration has occurred. Therefore, the performance of the baseline method is lower than the proposed method.

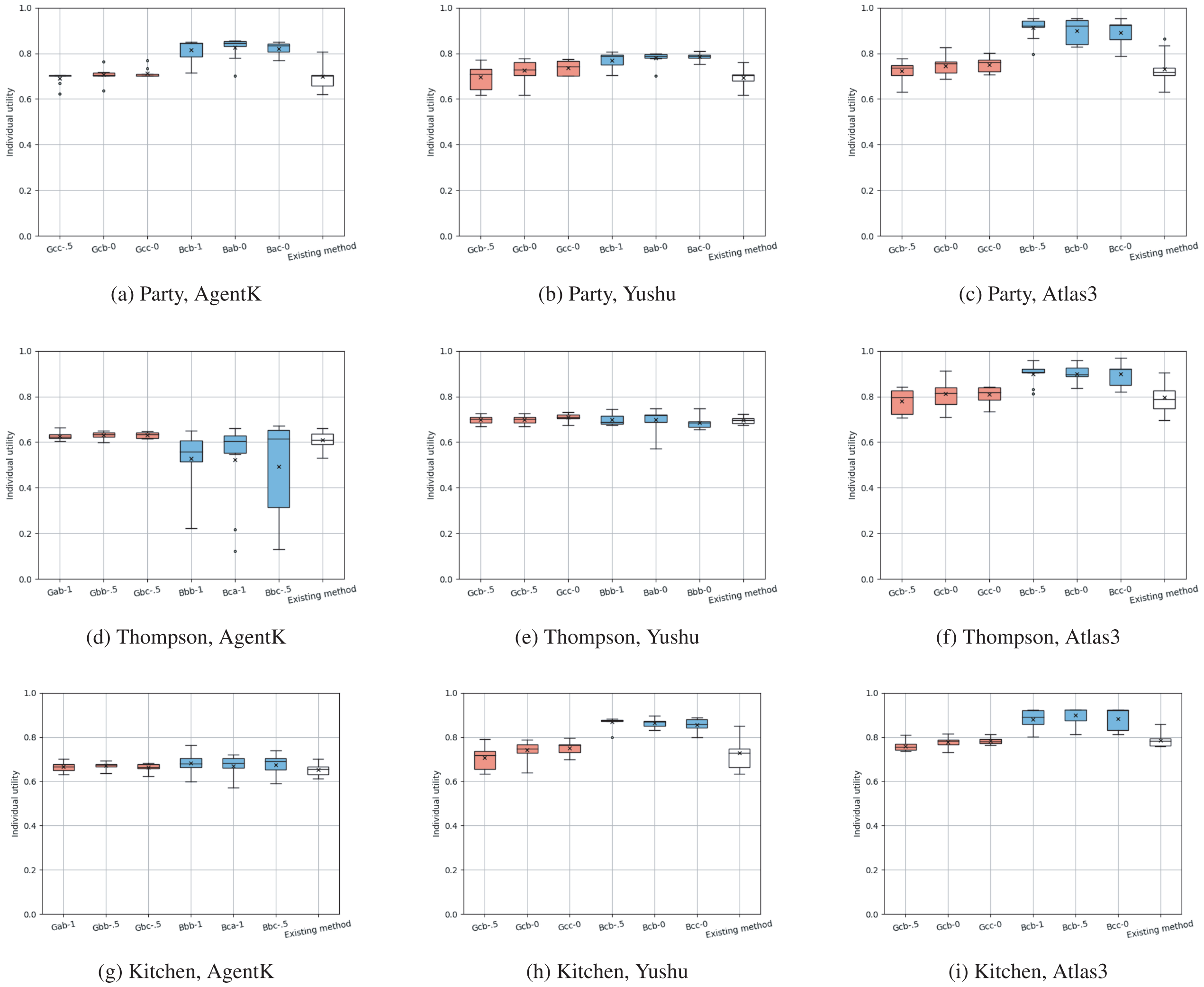
6.2.2 Performances under Various Domains and Agents

We evaluate our proposed approach under various domains or agents compared with the baseline, even if the domain or agent changes from the one in a training phase. In addition to the Party domain, the Thompson and Kitchen domains are used for negotiations. These domains were used in the experiments because they have different degrees of opposition from the England-Zimbabwe domain used in the training phase. In addition to AgentK used in Section 6.2.1, we use, as opponent agents, the Yushu [22] and the Atlas3 [6], which have an excellent record in ANAC.

For the combinations of agents and domains, the top three patterns of average utility values per agent used for learning and the existing methods are depicted in **Fig. 4** as box-and-whisker plots showing the distribution of average obtained utility values per 100 negotiation sessions. For negotiations in the Party domain, the results in Fig. 4 (a), Fig. 4 (b), and Fig. 4 (c) confirm that, as in Section 6.2.1, the averages are generally higher when learning with Boulware. In addition, Welch's t-test at a 1% level of significance confirmed that the patterns learned with Boulware were significantly different from the baseline.

In the Thompson domain, the results in Fig. 4 (d), Fig. 4 (e), and Fig. 4 (f) show that the results in Gahboninho are slightly higher than the baseline results when the opponent is AgentsK or Yushu. However, no significant differences were identified compared with the baseline results. On the other hand, as in the case of the Party domain, the average was higher when learning with Boulware and the opponent is Atlas3. Welch's t-test at the 1% significance level confirmed that the top three patterns were significantly different from the baseline.

In the Kitchen domain, the results in Fig. 4 (g), Fig. 4 (h), and Fig. 4 (i) show that both Gahboninho and Boulware have utility values similar to AgentK. This result is higher than the baseline, but there was no significant difference. On the other hand, as in the Party domain, the mean was higher when learned with Boul-



(a) Party, AgentK

(b) Party, Yushu

(c) Party, Atlas3

(d) Thompson, AgentK

(e) Thompson, Yushu

(f) Thompson, Atlas3

(g) Kitchen, AgentK

(h) Kitchen, Yushu

(i) Kitchen, Atlas3

Fig. 4 Box-and-whisker diagram for the utility value of negotiating in each domain. The labels in these figures indicate, in order, the initials of the opponent agent, Eq. (4), Eq. (5), and the reward for failure. For example, the case involving Gahboninho, Eq. (4-a), Eq. (5-b), and reward for failure -1 is represented by Gab–1, and the case involving Boulware, Eq. (4-c), Eq. (5-a), and reward for failure -0.5 is represented by Bca–.5.

ware and when the opponent is Yushu or Atlas3. The top three patterns were significantly different from the baseline by Welch's t-test at the 1% significance level.

The overall trend is that Eqs. (4-c) and (5-b) are frequently used in the best three patterns. Figure 4 shows that the best three patterns learned in Gahboninho do not change significantly in value in any of the domains. Conversely, the best three patterns learned by Boulware are relatively higher in all domains except the Thompson domain, but are not very stable in the Thompson domain, indicating that they vary widely from domain to domain. The reason why negotiations in the Thompson domain resulted in a different trend than in the other two domains is the high degree of opposition. In domains with a high degree of opposition, there are few bids in which both the utility of the opponent and its utility are high, making it difficult for the opponent to make a bid that is advantageous to us, and it is difficult to reach an agreement. Therefore, in training with Boulware, the agents tend to wait for a better opponent's bid, which did not perform so well. On the other hand, the negotiations with Atlas3 showed the same tendencies as the others, probably because the bidding strategy in Atlas3 is relatively easy to make a more advantageous bid to us.

7. Conclusion

We propose a new configuration of a deep reinforcement learning framework for automated negotiation acceptance strategies using DQN. We also evaluated reward functions that are effective in learning acceptance strategies, and conducted experiments to compare the reward functions that can obtain higher utility values. The experimental results showed that the utility values were significantly higher than those of the existing methods for several combinations of rewards trained by the Boulware agent.

One topic for future work is evaluating the opponent agent's acceptance strategy. All experiments in this paper were conducted with the opponent agent's acceptance strategy disabled to obtain accurate learning results. However, it is difficult to predict that the opponent would not accept our bid at all in an original negotiation. Therefore, it is necessary to consider the bid to the opponent agent.

Another future direction is to consider domains with a high degree of opposition. The experiments in domains with a high degree of opposition did not yield better results than negotiations in other domains. Therefore, it is necessary to consider bidding strategies that take into account the acceptance of the opponent agent.

In this paper, only two elements were extracted from the baseline and used as input elements because these were deemed particularly important. However, it can be assumed that some of the opponent's bidding strategies are influenced by one's bid. In addition, if the opponent's acceptance strategy exists, the utility value may be higher if the opponent accepts the bid rather than accepting it himself. Furthermore, as mentioned above, there may be cases where a reservation value or discount factor is set. Therefore, we can expect improved performance by including as input elements the utility value that one intends to propose next, or a utility value that considers the discount factor.

Acknowledgments This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers 22H03641, 19H04216 and JST FOREST (Fusion Oriented REsearch for disruptive Science and Technology) Grant Number JPMJFR216S.

References

[1] Fiedler, A. and Sackmann, D.: Automated negotiation for supply chain finance, Mes, M., Lalla-Ruiz, E. and Voß, S. (Eds.), *Computational Logistics*, pp.130–141, Springer International Publishing (2021).

[2] Etukudor, C., Couraud, B., Robu, V., Früh, W.-G., Flynn, D. and Okereke, C.: Automated negotiation for peer-to-peer electricity trading in local energy markets, *Energies*, Vol.13, No.4, p.920 (2020).

[3] Chater, N., Misyak, J., Watson, D., Griffiths, N. and Mouzakitis, A.: Negotiating the traffic: Can cognitive science help make autonomous vehicles a reality?, *Trends in Cognitive Sciences*, Vol.22, No.2, pp.93–95 (2018).

[4] Aydoğan, R., Baarslag, T., Fujita, K. and Jonker, C.: *The 13th International Automated Negotiating Agents Competition* (*ANAC2022*) (2022), available from ⟨https://web.tuat.ac.jp/˜katfuji/ANAC2022/⟩.

[5] Kawaguchi, S., Fujita, K. and Ito, T.: Agentk: Compromising strategy based on estimated maximum utility for automated negotiating agents, *New Trends in Agent-Based Complex Automated Negotiations*, pp.137–144, Springer (2012).

[6] Mori, A. and Ito, T.: Atlas3: A negotiating agent based on expecting lower limit of concession function, *Modern Approaches to Agent-based Complex Automated Negotiation*, pp.169–173, Springer (2017).

[7] Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K. and Jonker, C.M.: Genius: An integrated environment for supporting the design of generic automated negotiators, *Computational Intelligence*, Vol.30, No.1, pp.48–70 (2014).

[8] Sunder, V., Vig, L., Chatterjee, A. and Shroff, G.: Prosocial or selfish? agents with different behaviors for contract negotiation using reinforcement learning, *International Workshop on Agent-Based Complex Automated Negotiation*, pp.63–81, Springer (2018).

[9] Bakker, J., Hammond, A., Bloembergen, D. and Baarslag, T.: Rlboa: A modular reinforcement learning framework for autonomous negotiating agents, *Proc. 18th International Conference on Autonomous Agents and Multiagent Systems*, pp.260–268 (2019).

[10] Takahashi, T., Higa, R., Fujita, K. and Nakadai, S.: Venas: Versatile negotiating agent strategy via deep reinforcement learning, *Thirty-Sixth AAAI Conference on Artificial Intelligence* (*AAAI 2022*), pp.13065–13066, AAAI Press (2022).

[11] Higa, R., Fujita, K., Takahashi, T., Shimizu, T. and Nakadai, S.: Reward-based negotiating agent strategies, *Thirty-Seventh AAAI Conference on Artificial Intelligence* (*AAAI 2023*). AAAI Press (2023).

[12] Baarslag, T., Hindriks, K. and Jonker, C.: Effective acceptance conditions in real-time automated negotiation, *Decision Support Systems*, Vol.60, pp.68–77 (2014).

[13] Razeghi, Y., Yavuz, C.O.B. and Aydoğan, R.: Deep reinforcement learning for acceptance strategy in bilateral negotiations, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol.28, No.4, pp.1824–1840 (2020).

[14] Baarslag, T., Aydoğan, R., Hindriks, K.V., Fujita, K., Ito, T. and Jonker, C.M.: The automated negotiating agents competition, 2010–2015, *AI Magazine*, Vol.36, No.4, pp.115–118 (2015).

[15] Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A. and Jonker, C.: Decoupling negotiating agents to explore the space of negotiation strategies, *Novel insights in Agent-based Complex Automated Negotiation*, pp.61–83, Springer (2014).

[16] Sengupta, A., Nakadai, S. and Mohammad, Y.: Transfer learning based adaptive automated negotiating agent framework, *Proc. 31st International Joint Conference on Artificial Intelligence, IJCAI-22*, pp.468–474, International Joint Conferences on Artificial Intelligence Organization (2022).

[17] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol.9, No.8, pp.1735–1780 (1997).

[18] Rubinstein, A.: Perfect equilibrium in a bargaining model, *Econometrica: Journal of the Econometric Society*, pp.97–109 (1982).

[19] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W.: Openai gym, *arXiv preprint arXiv:1606.01540* (2016).

[20] Mohammad, Y., Nakadai, S. and Greenwald, A.: Negmas: A platform for automated negotiations, *International Conference on Principles and Practice of Multi-Agent Systems*, pp.343–351, Springer (2020).

[21] Adar, M.B., Sofy, N. and Elimelech, A.: Gahboninho: Strategy for balancing pressure and compromise in automated negotiation, *Complex Automated Negotiations: Theories, Models, and Software Competitions*, pp.205–208, Springer (2013).

[22] An, B. and Lesser, V.: Yushu: A heuristic-based agent for automated negotiating competition, *New Trends in Agent-Based Complex Automated Negotiations*, pp.145–149, Springer (2012).

Hyuga Matsuo received a B.E. degree from the Tokyo University of Agriculture and Technology in 2022. He is currently a student in the master's course at Tokyo University of Agriculture and Technology. He is interested in automated negotiation.

Katsuhide Fujita is a Professor at Institute of Global Innovation Research, Tokyo University of Agriculture and Technology. He received his B.E., M.E, and Doctorate of Engineering from Nagoya Institute of Technology in 2008, 2010, and 2011, respectively. From 2010 to 2011, he was a research fellow of Japan Society for the Promotion of Science (JSPS). During 2010 and 2011, he was a visiting researcher at MIT Sloan School of Management. From 2011 to 2012, he was a Project Researcher at School of Engineering, The University of Tokyo. He was an Associate Professor at Institute of Engineering, Tokyo University of Agriculture and Technology from 2012 to 2023. Since 2023, he has been a Professor at Institute of Global Innovation Research, Tokyo University of Agriculture and Technology. His main research interests include multi-agent systems and text mining.