Reward-Based Negotiating Agent Strategies

Ryota Higa[1,2], Katsuhide Fujita[2,3], Toki Takahashi[2,3], Takumu Shimizu[2,3], Shinji Nakadai[1,2]

[1]NEC Corporation, Japan
[2]National Institute of Advanced Industrial Science and Technology(AIST), Japan
[3]Tokyo University of Agriculture and Technology, Japan
r-higaryouta@nec.com, katfuji@cc.tuat.ac.jp, {takahashi, shimizu}@katfuji.lab.tuat.ac.jp, nakadai@nec.com

Abstract

This study proposed a novel reward-based negotiating agent strategy using an issue-based represented deep policy network. We compared the negotiation strategies with reinforcement learning (RL) by the tournaments toward heuristics-based champion agents in multi-issue negotiation. A bilateral multi-issue negotiation in which the two agents exchange offers in turn was considered. Existing RL architectures for a negotiation strategy incorporate rich utility function that provides concrete information even though the rewards of RL are considered as generalized signals in practice. Additionally, in existing reinforcement learning architectures for negotiation strategies, both the issue-based representations of the negotiation problems and the policy network to improve the scalability of negotiation domains are yet to be considered. This study proposed a novel reward-based negotiation strategy through deep RL by considering an issue-based represented deep policy network for multi-issue negotiation. Comparative studies analyzed the significant properties of negotiation strategies with RL. The results revealed that the policy-based learning agents with issue-based representations achieved comparable or higher utility than the state-of-the-art baselines with RL and heuristics, especially in the large-sized domains. Additionally, negotiation strategies with RL based on the policy network can achieve agreements by effectively using each step.

Introduction

Negotiation is essential for establishing cooperation and collaborations in multi-agent systems. Automated negotiation has been used in various fields, including game theory, artificial intelligence, and social sciences (Kraus 2001; Jennings et al. 2001; Fatima, Kraus, and Wooldridge 2014). These negotiation agents are used to assist humans in various critical tasks. These strategies are applied to collaborate in common real-world cases, such as e-markets (Bagga et al. 2021) and cooperative behavior among robots (Inotsume et al. 2020). In the future, automated negotiators can support trades among companies in the real world and be used to construct effective and flexible supply chain networks (Mohammad et al. 2019). Therefore, the development of automated negotiating agents for is critical.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Agent strategies utilizing reinforcement learning (RL) have attracted considerable research attention in bilateral multi-issue negotiation because of their ability to adapt to various scenarios and opponents (Bakker et al. 2019; Bagga et al. 2020). According to existing RL research, *the reward is sufficient to drive behavior that exhibits abilities studied in natural and artificial intelligence, including knowledge, learning, perception, social intelligence, language, generalization and imitation* (Silver et al. 2021). Existing studies on negotiation architecture using RL incorporate the utility function that is rich and provides concrete information. The rewards of RL are considered as more generalized signals in practice. Thus, when the utility of agents is uncertain or ordinal, the reward-based negotiating agent strategy using RL is acceptable. This study focused on reward-based negotiation agent strategies by applying an end-to-end RL model without the heuristic components requiring expert knowledge, experiments, or utility functions (Baarslag et al. 2014). Thus, acceptance and bidding strategies can be learned without decoupling the negotiation strategy and concrete utility information.

Versatile negotiating agent strategy (VeNAS) through deep RL (Takahashi et al. 2022) is a data-driven negotiation strategy in which the reward function is defined by the utility function. However, the limitation of VeNAS is that the performance under the negotiation with large-sized domains decreases with the increase in the size of the output space. This phenomenon can be attributed to the fact that most learned negotiation strategies based on VeNAS cannot reach an agreement with the opponents in the training phase. Furthermore, the properties of the reward-based negotiation strategy are yet to be comprehensively investigated.

This study proposed a reward-based negotiating agent strategy through a multi-issue policy network. The policy network, which is a fairly neural network (NN) architecture, was trained to predict the optimal policy in policy-based RL without incorporating utility functions. The characteristics of multi-issue negotiation domains were considered to improve scalability to the domain size. Comparative studies have analyzed the significant properties of negotiation strategies with RL. We detailed that policy-based learning agents with issue-based representations achieve comparable or higher utility than the state-of-the-art baselines with RL and heuristics in large-sized domains.

The contributions of this study are as follows:

- A novel reward-based learning agent architecture was proposed. This strategy is an end-to-end deep RL in which the input is the opponent's offer and output is the next action of the agent, including accepting or offering the bid in a bilateral multi-issue negotiation, not incorporating the utility function.
- The Markov decision process (MDP) and multi-issue policy network model were defined for bilateral negotiations. Issue-based state and action representations are key for achieving the multi-issue policy network model.
- The significant properties of negotiation strategies were clarified with deep RL by performing comparative experiments. The proposed policy-based learning agents with issue-based representations achieved comparable or higher individual utility, agreement rates, and social welfare than the state-of-the-art baselines with RL and heuristics. In negotiation strategies with RL based on the policy network, agreements are achieved using drastic small number of steps and improving step efficiency.

Related Work

Automated negotiation strategies have attracted considerable research attention. Motivated by the challenges of bilateral negotiations between automated agents, the automated negotiating agents competition (ANAC) was organized in 2010 (Baarslag et al. 2015). The evolution of strategies and critical factors for developing the competition have been proved by analyzing the results from the ANAC.

This model identified three components that constitute a negotiation strategy, namely the bidding strategy, opponent model, and acceptance strategy (Baarslag et al. 2014). In this architecture, the negotiation performance of selecting advanced BOA techniques were compared to provide an overview of the factors influencing the final performance. Particularly, well-performing opponent modeling techniques have attracted considerable attention in the field (Baarslag et al. 2016). Moreover, effective strategies can be achieved by combining the modules of superior agent's strategies in competitions, depending on the opponent's strategies and negotiation environments. Several sophisticated existing agent strategies comprise a fixed set of modules. Therefore, studies on negotiation strategies that focus on the modules are crucial.

Studies have focused on the divided component parts of automated negotiating strategies, including primary bidding, acceptance, and opponent modeling. Effective strategies can be achieved by combining the modules of top agents' strategies in these competitions, depending on the opponents' strategies and negotiation environments. Several sophisticated agents' strategies that currently exist comprise a fixed set of modules (Ilany and Gal 2016; Kawata and Fujita 2019). Sengupta et al. (Sengupta, Mohammad, and Nakadai 2021) proposed a mechanism for selecting effective strategies using maximum entropy RL by using a deep-learning-based opponent classifier.

Negotiation strategies have been used in the RL approach of Q-learning for bidding (Bakker et al. 2019), and deep Q-learning (DQN) was used for acceptance (Razeghi, Yavus, and Aydoğan 2020). These studies have focused on estimating the threshold target utility for bidding and acceptance, that is, on the divided parts of negotiating strategies. Bagga et al. (Bagga et al. 2020) used a deep deterministic policy gradient algorithm to negotiate with multiple sellers concurrently in electronic markets. However, their motivation differed considerably from the reward-based negotiating agent strategies using RL; their learning model included neither bidding nor only packaged actions for concurrent negotiations. Additionally, these models incorporate the utility function that is rich and contains certain information for the negotiation in learning the negotiation actions (bidding, acceptance, and opponent modeling).

Negotiation Environment

A bilateral multi-issue negotiation with two agents, A and B, was assumed for a negotiation domain D, which defines a set of issues $I = \{I_1, I_2, \ldots, I_n\}$ and possible values $V_i = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$, where n is the number of issues, and k_i is the number of values for issue I_i. A set of values selected for each issue is referred to as an outcome. Each agent proposes a bid during a negotiation. Here, Ω is the set of all possible outcomes, and it is a common knowledge shared among all agents in a negotiation session. Every agent has a unique preference profile that represents its own preferences for outcome $\omega \in \Omega$. The utility of an outcome is defined by a utility function $U(\cdot)$, which is normalized to a real number in the range $[0, 1]$. The utility function of each agent is not shared with other agents

A negotiation session has a timeline t, represented as a real number in the range $[0, T]$, where T is a deadline. Here, $t = 0$ refers to the starting time of a negotiation, and $t = T$ represents the deadline. In turn-based games, the game flow is categorized into defined parts, namely steps or turns. When a player takes an action, the number of steps/turns increases by one. When every player takes their steps/turns, the round is over, and the number of rounds is increased by one.

Alternating Offers Protocol. The interaction between negotiating parties is regulated by negotiation protocol P, which defines a set of rules that formalize how and when proposals can be exchanged. Here, P is agreed before the agents start negotiation. The alternating offers protocol (AOP), which is a bilateral negotiation protocol(Rubinstein 1982), was considered. In AOP, the negotiating parties exchange offers in turns and each agent has three possible actions as follows: *Accept, Offer, EndNegotiation*. If the action is an *Offer*, agent *X* is subsequently asked to determine its next action; the agents continue to take turns into the next round. If it is not an *Offer*, the negotiation is over. The final score is determined for each agent when an offer ω is accepted by the opponent at time t; each agent obtains $U(\omega)$. When the action is returned as an *EndNegotiation*, each agent obtains 0. If a negotiation is not concluded within the deadline ($t = T$), each agent obtains 0[1].

[1]In this study, the reservation value is assumed as 0

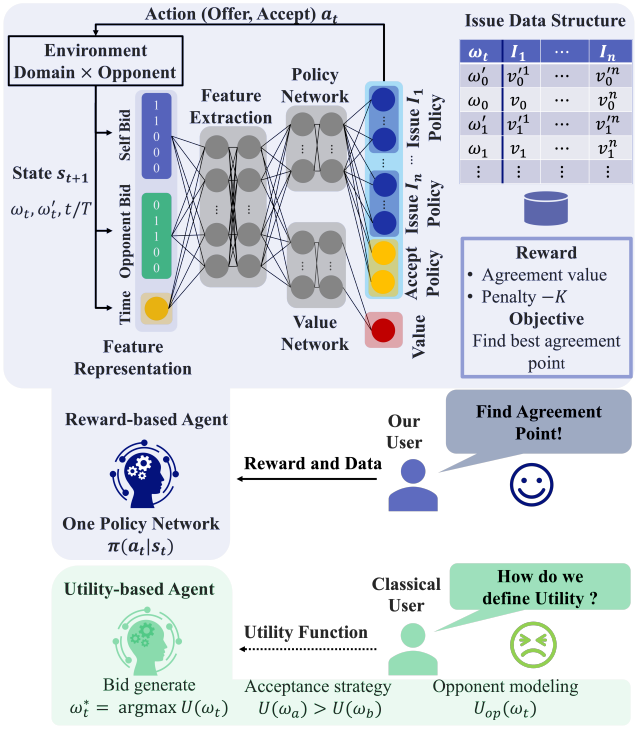


Figure 1: Reward-based negotiating agent strategy. The input of the learnable agent is the bid that is vectorized by one-hot encoding for each issue and time defined as states. The nodes of the output layer are allocated vectorized bids grouping as multi-issue for offer, acceptance, and value. The hidden layer consists of an end-to-end learning model based on a policy network. Unlike the existing utility-based method in the bottom figure, users are not required to prepare a utility function, generate a bid by the argmax operation on the utility function, or make an agreement decision. Modeling of the utility of the opponent is not required. Have the agents learn through trial and error the policies that will lead to an excellent final point of agreement.

Reward-Based Negotiating Agent Strategy

Figure 1 illustrates the proposed RL architecture through a multi-issue policy network and compares existing utility-based methods. The environment includes the history of the bids exchanged between the agent and the opponent. The opponent has its utility function and strategy; however, these are unknown to the agent. Therefore, the proposed learning agent does not include them as the input, reward, and body of the agent. To avoid information loss of the domain and use the raw bid directly as the state, the bid is vectorized by one-hot encoding for each issue. The nodes of the input layer are allocated as vectorized bids and time defined as states. The nodes of the output layer are allocated vectorized bids grouping as multi-issue for offer, acceptance, and value. A part of the hidden layer consists of an end-to-end learning model based on a policy network.

Compared with previous studies((Bagga et al. 2020) etc.), the action a_t of the proposed architecture covers all negotiation actions (*Offer* and *Accept*). The input is the bid that is not the bid's utility calculated by the utility function; the output is the negotiation action, including the offer and its bid and acceptance. The body of the agent does not include its own utility function for learning the negotiation strategy, in which the proposed architecture utilizes its own utility function to obtain the reward only. Therefore, the proposed architecture achieves comprehensive agent's architecture in a learning framework, which is end-to-end RL for the negotiating the strategy of the agent.

MDP for Bilateral Multi-Issue Negotiation

To apply machine learning to negotiation agents using direct negotiation bid data without exact utility function and the historical data of the utility, an MDP was formulated for bilateral multi-issue negotiations. A finite MDP is provided as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, p(s_{t+1}|s_t, a_t), p(s_0) \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the set of rewards, $p(s_{t+1}|s_t, a_t)$ is the transition function, and $p(s_0)$ is the initial state distribution. The policy base probability density function of the trajectory data on the first-order MDP is given as $p(s_0, a_0, \cdots, a_T, s_{T+1}) = p(s_0) \prod_{t=0}^{T} p(s_{t+1}|s_t, a_t) \pi_\theta(a_t|s_t)$, where $\pi_\theta(a_t|s_t)$ is the policy function that generates the actions of agents. An AOP is defined using a finite MDP as follows:

Time step $t \in \{0, 1, \cdots, T\}$: Turn of negotiation from the initial time to the negotiation deadline T.

State set of state $s_t \in \mathcal{S}$: The agent's offer ω_t, opponent's offer ω'_t and accept signal η'_t, and t/T.

Action set of action $a_t \in \mathcal{A}$: The agent's selected offer ω_t and accept signal η_t. The action space is proportional to the domain size of the utility function.

Reward $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$: When the agent accepts, the agent obtains the final utility value. Penalty $-K$ is given when the negotiation ends without reaching an agreement. Otherwise, the reward is 0.

Policy function $\mathcal{S} \times \mathcal{A} \to [0, 1]$: The policy function is defined as $\pi_\theta(a_t \mid s_t) := \Pr(A_t = a_t \mid S_t = s_t, \theta_t = \theta)$.

Transition function $\mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$: Transition function is defined as $p(s_{t+1}|s_t, a_t) := \Pr(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t)$.

History $\mathcal{D} = (\omega_0, \omega'_0, \cdots, \omega_t, \omega'_t)$: The observable data during a negotiation.

A method for learning the measure function and proposal for an issue-based model is presented as follows:

Multi-Issue Policy Network and Learning

Policy Gradient Algorithms. Policy base objective function $J(\theta)$ to learn parameters θ is defined as follows:

$$J(\theta) := \mathbb{E}_\pi \left[\sum_{t=0}^{T} r(s_t, a_t) \middle| S_0 = s_0, \pi_\theta \right],$$

where, θ is updated by $\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta)$. By the policy gradient (PG) theorem(Sutton et al. 1999), the gradient $\nabla_\theta J(\theta)$ is calculated semi-analytically as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi \left[\nabla_\theta \ln \pi_\theta(a \mid s) Q(s, a) \right].$$

For one of the extensions of PG, we used the actor-critic methods in which policy and value functions are trained simultaneously. A brief notation is as follows:

$$\begin{aligned}\nabla_\theta J(\theta) &:= \mathbb{E}_\pi\left[\nabla_\theta \ln \pi_\theta(a \mid s) A_\phi(s,a)\right], \\ A_\phi(s,a) &:= Q_\phi(s,a) - V_\phi(s), \\ Q_\phi(s,a) &:= \mathbb{E}_\pi\left[r(s,a) + V_\phi(s') | S_0 = s, A_0 = a\right], \\ V_\phi(s) &:= \mathbb{E}_\pi\left[\sum_{t=0}^{T} r(s_t, a_t) \middle| S_0 = s\right].\end{aligned}$$

The Q_ϕ and V_ϕ are approximated by the NN and optimized to minimize the mean square error (MSE) from cumulative rewards. Furthermore, proximal policy optimization (PPO), which is a stable and well-known policy gradient method, was applied to optimize learning parameters (Schulman et al. 2017).

Multi-Issue Policy Network. A deep strategy network model was proposed for combinatorial issue and shared policies were accepted as follows:

$$\pi_\theta(a|s) := \pi_{\theta_A}(a_A|h(s)) \prod_{i=I_1}^{I_n} \pi_{\theta_i}(a_i|h(s)),$$

where $\eta \sim \pi_{\theta_A}(a_A|h(s))$ is the accept strategy policy function that generates an accept signal $\eta \in \{0,1\}$, $v^i_{k_i} \sim \pi_{\theta_i}(a_i|h(s))$ is the issue I_i's policy function that generates possible values. Here, $v^i_{k_i}$, and $h(s)$ are feature and hidden parameters to input policy functions. This study was inspired by the action branching architecture (Tavakoli, Pardo, and Kormushev 2018) and discrete and continuous action approaches (Delalleau et al. 2019). PG loss functions can be calculated independently, as in $\ln \pi_\theta(a|s) = \ln \pi_{\theta_A}(a_A|h(s)) + \sum_{i=I_1}^{I_n} \ln \pi_{\theta_i}(a_i|h(s))$, and this corresponds to scalable when applied to large issue spaces. The details of the NN structure and the hyperparameter are presented in Table 1.

Issue-Based State Representation. As a representation of the general state representation vector, we define the function $Vec(\cdot)$, $Vec(\omega_t, t) := (Vec(v^1), \cdots, Vec(v^n), t/T)$.

A concrete example of $Vec(v^i)$ is as follows: Categorical $v \in V_i$ is one-hot encoding (example, food), Ordinal $v \in V_i$ is normalization and standardization (example, price), Continuous $v \in V_i$ is normalization and standardization (example, time). In the future, the same mechanism can be used by using human facial expressions or voice data by generating features with the CNN or other methods.

Issue-Based Action Representation. To select action ω, the RL architecture computes $\omega_t = \arg\max_{a_t} Q(s_t, a_t)$, or samples by policy-method $\omega_t \sim \pi_\theta(a_t|s_t)$.

In the proposed method, the issue data structure was used for action generation, $\omega_t = (v^1, \cdots, v^n)$. Offer is the tuple of issues, and each value of issue is sampled from each policy model $v^i \sim \pi_{\theta_i}(a_i|h(s))$. Additionally, this method can be applied to various negotiation strategy models by applying $softmax$, if I_n is discrete, or Gaussian, if I_n is a continuous number. Therefore, handling numerous variables is critical because of applications from a game-like evaluation environment to a realistic negotiation domain.

Experiments and Evaluations

Some packages to realize the proposed idea are available; the following evaluations were made by improving the package for *NegMAS* platform(Mohammad et al. 2019) and RL-baselines(Raffin et al. 2021).

Negotiation Environment Settings

We focused on the essential repeated encounter bilateral multi-issue negotiation of using the same domains and opponents as those adopted for training. Numerous advanced techniques beyond the scope of this study are required to improve negotiation setting that involves working effectively in a different domain and opponent from that of the training (example, transfer-learning, meta-learning).

The deadline (T) is set to 40 rounds, and in this setting, the negotiation ends when both agents have each acted 40 times. This deadline is determined based on the settings in existing reference (Bagga et al. 2020). If T is set as the continuous time or other values, all results are satisfactory by running several iterations to obtain sufficient training.

Utility Function. In these experiments, the weighted-sum utility function case that w_i ($\sum_{i=1}^{n} w_i = 1.0$) be the weight of each issue I_i is considered. The weighted-sum utility function is most popular settings in automated negotiation (Baarslag et al. 2015). The utility $U(\cdot)$ of the outcome ω is $U(\omega) = \sum_{i=1}^{n} w_i \cdot e_i(\omega)$, where $e_i(\omega)$ is the evaluation value of the option for issue I_i of ω, normalized to range$[0, 1]$. w_i and $e_i(\cdot)$ are defined in the agent profile.

Domains. To prove that the proposed approach can be trained in various negotiation domains, we consider domains with comprehensive large sizes of outcome space $|\Omega|$ (from 10 to 10^4 domain sizes) and oppositions. The sizes and oppositions of all domains are in Table 2[2].

Opponents. To detail the adaptability to various opponents, two types of the basic negotiating agents were considered based on the categorization of strategies (Faratin, Sierra, and Jennings 1998) and the champions of the previous ANACs (*AgentK, HardHeaded, Atlas3*, and *AgentGG*). The basic bidding strategies of the agents are three time-dependent strategies (*Boulware*, *Linear*, and *Conceder*), and two behavior-dependent strategies (*Tit-For-Tat1* and *Tit-For-Tat2*). Their acceptance condition is based on $AC_{next}(\omega_t)$ that it accepts the offer of the opponents when its utility is higher than the utility of the counteroffer of the agents. We selected suitable champions under defined negotiation environments from the previous competitions, including the opponent modeling and heuristic techniques.

Experimental Settings

The performance of the agents was scored by their obtained utility under a deterministic policy and evaluated based on the average scores, out of 100 negotiations. To stabilize learning, we trained with 10 initial values and adopted the learning that exhibited the best performance.

[2]These negotiation domains are included in the negotiation platform Genius (Lin et al. 2014).

The code was implemented in Python 3.8 and run on 28 core CPUs with 128 GB of memory with Ubuntu Desktop 22.04 as the operating system. The total experimental period was approximately three months because of the enormous negotiation tournaments, including $7(domains) \times 9(oppositions) \times 4(approaches)$ kinds of tournaments in training and test phases.

Comparison. For implementing the proposed architecture, the PPO was applied (Schulman et al. 2017). The PPO is a typical policy gradient method for RL that alternates between sampling data through interaction with the environment. In the model, a surrogate objective function is optimized using stochastic gradient ascent by enabling multiple epochs of minibatch updates.

Reward-Based Models Utilizing RL

- MiPN: Proposed architecture including a multi-issue policy network with the PPO.
- PPO-VeNAS: The VeNAS including policy network not considering the multi-issue action representations with PPO. To measure the performance of VeNAS applying the PPO, the VeNAS-agent was trained for utilizing the PPO.
- VeNAS(DQN): The original VeNAS including Deep Q-Network (DQN) not considering the multi-issue action representations. To compare with the performance of VeNAS applying the PPO, the VeNAS-agent was trained for utilizing the DDQN.
  Data or bid-based state: $s_t := \{\omega_{t-1}, \omega'_{t-1}, \omega_t, \omega'_t, t/T\}$.

Utility-Based Models with the Domain Knowledge

- DRBOA: We developed DRBOA (deep RLBOA) based on a state-of-the-art negotiation agent that utilizes RL (Bakker et al. 2019). To reduce the difference of the learning methods among architectures, the original RLBOA-agent was developed to be trained by the DDQN. The model trained with the DDQN outperformed the original RLBOA with Q-learning. The acceptance condition is based on $AC_{next}(\omega_t)$ that it accepts the offer of the opponents when its utility is higher than the utility of the counteroffer of the agent.
  Domain knowledge or utility-based state:
  $s_t := \{U(\omega_{t-1}), U(\omega'_{t-1}), U(\omega_t), U(\omega'_t), t/T\}$.
- Heuristics: This score is the baseline referring to the average of scores that each opponent agent negotiates with other opponents except for itself.

Reward. In the experiments, the final utility values in making agreements are used as the reward functions of MiPN, PPO-VeNAS, VeNAS(DQN). When the agent accepts, $r(\{..., \omega'_t\}, \eta_{t+1}) = U(\omega'_t)$ is rewarded. When the opponent accepts, $r(\{..., \omega_t, \eta'_{t+1}\}, \omega_t) = U(\omega_t)$ is rewarded. Penalty $-K$ is given when the negotiation ends without reaching an agreement. Otherwise, the reward is 0. $K = 1$ was set for failure to reach an agreement in the experiments.

Hyperparameter	Value
Horizon (T)	2048
Adam stepsize	3×10^{-4}
Num. epochs	10
Minibatch size	64
Discount (γ)	0.99
GAE parameter (λ)	0.95
Clipping parameter (ϵ)	0.2
Feature Extractor	Shared
Shared Network	Flatten
Activation Function	Tanh
Value fc layers	[64, 64]
Policy fc layers	[64, 64]
Policy Final Layer	Issue-decoupled $Softmax$
Value Loss	MSE
Policy Loss	Multicategorical Cross Entropy
Optimizer	Adam Optimizer

Table 1: Hyperparameters of the NN for training.

The body of the reward-based models (MiPN, PPO-VeNAS, VeNAS(DQN)) does not include the utility function for learning the negotiation strategy. Their architectures incorporate the utility function to obtain the reward only. Therefore, reward functions of these models do not stick to the utility function. The utility-based models should incorporate the utility function to work well. Therefore, the reward functions of the reward-based models are used as the utility function to evaluate the performances fairly in the experiments.

Hyperparameters. The training period was 500,000 steps. As a policy network, a NN with two hidden layers of 64 units, and a $tanh$ function was used as the activation function. The outputs are the probability of proposing the value for each issue and the probability of accepting the offer. The detailed hyperparameters are provided in Table 1.

Experimental Results

Performance (Individual Utility). Table 2 and Figure 2(a) reveal the individual utilities toward nine opponents among the state-of-the-art approaches with RL and Heuristics under seven domains with various sizes and oppositions. The RL-based approaches exhibit higher utilities than Heuristics. Therefore, RL-based approaches can obtain the strategy that is more adaptive to the environment than the heuristic strategy. Especially, the agents trained by the proposed architecture with a multi-issue policy network (MiPN) achieve comparable or higher utilities than other approaches. Thus, MiPN can learn and adapt various negotiation strategies using the proposed learning architecture without designing an effective strategy that considers the strategies and domains of the opponents.

VeNAS(DQN) drastically decreased the individual utilities as the domain size was large because the several trained agents using the DDQN could not achieve agreements in large-sized domains. Without considering the proposed multi-issue policy network, the scalability of the architecture was not sufficient in large-sized domains.

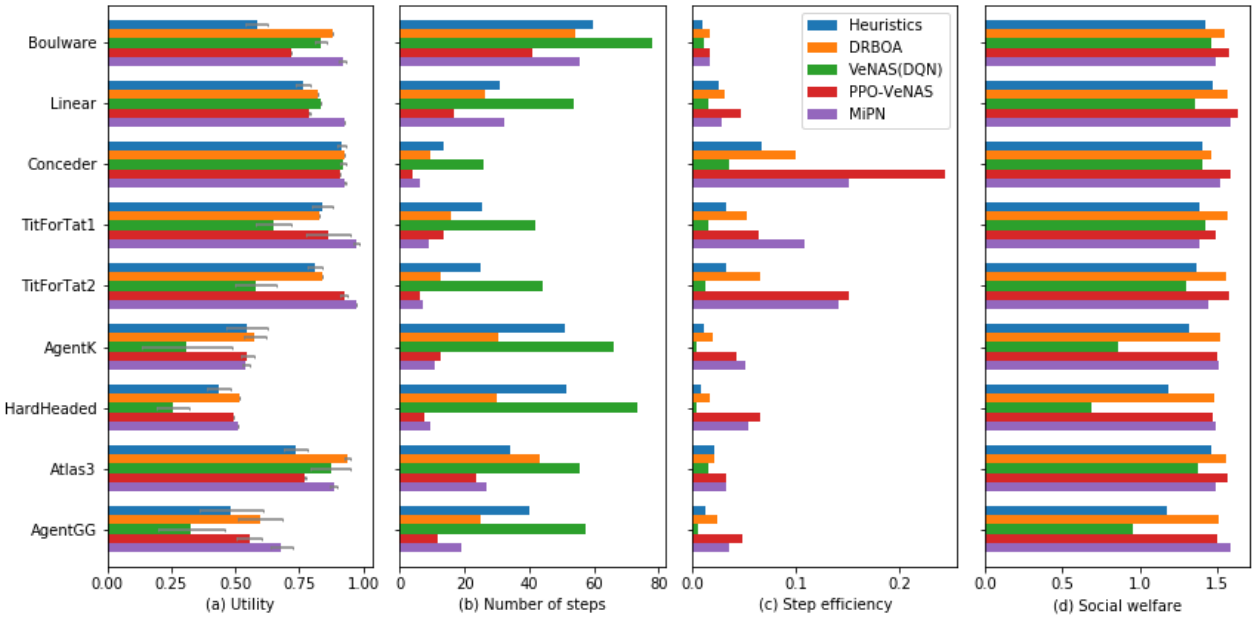


Figure 2: (a) Individual utility toward each opponent among five comparative approaches. (b) Number of steps toward each opponent among five comparative approaches. (c) Step efficiency (*(Individual utility)/(the number of steps)*) toward each opponent among five comparative approaches. (d) Social welfare toward each opponent among five comparative approaches. The variance of the results is omitted because it was small.

DRBOA exhibited second- or third-highest utility because the size of the state and action space of DRBOA were maintained constant by considering the utility function even when the domain size increased, whereas VeNAS increased the size of the state and action space when the domain size increased.

The RL-based architectures exhibited weak performance for *AgentK* and *Hardheaded* because *AgentK* exhibits randomness and exploits specific opponents. Thus, these architectures cannot effectively address behaviors not observed in the training phase. Additionally, *Hardheaded* does not consider the risk of disagreements but continue to make a concession considering the risk of disagreements.

Performance (Step Efficiency). Figure 2(b) and (c) present the number of steps and the step efficiency toward each opponent. MiPN and PPO-VeNAS exhibit higher step efficiencies than other approaches because in these models, reach agreements are achieved in shorter steps. This characteristic is caused by the policy-based approach; therefore, PPO-VeNAS tends to make agreements in less time. By contrast, MiPN ensures superior balances between the number of steps and individual utilities than PPO-VeNAS. Comparing RL-based approaches with Heuristics, RL-based approaches except for VeNAS(DQN) can determine superior agreements in the limited number of steps because Heuristics extends the number of steps in the negotiation to obtain more utility information of the opponent. However, RL-based approaches lose utilities toward *Boulware* because these methods make agreements in the shorter steps without extending the negotiation steps.

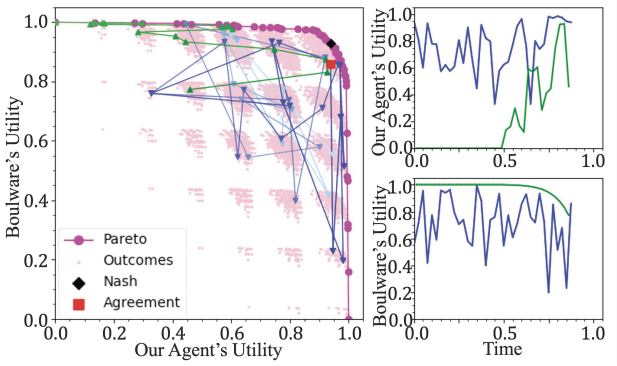
Performance (Social Welfare). Figure 2(d) presents the average social welfare toward each opponent. In addition to selfishness, the cooperativeness of the strategy and sociability can be evaluated. The results confirmed that the social welfare of MiPN, PPO-VeNAS, and DRBOA are higher than that of other approaches. Especially, MiPN and PPO-VeNAS achieved the highest social welfare despite not considering the social welfare in their architecture. By altering the reward function to consider the opponent's utility, a higher social welfare can be obtained easily. DRBOA also obtains the highest social welfare in the tournaments toward some opponents because DRBOA considers the estimated opponent's utilities in determining the next offer, unlike VeNAS and MiPN.

Performance (Agreement Rate). The total agreement rate of MiPN, PPO-VeNAS, VeNAS(DQN), Heuristics, and DRBOA were 99.9%, 99.3%, 84.0%, 93.0%, 99.7%, respectively. MiPN and PPO-VeNAS exhibited higher agreement rates than Heuristics because they can make agreements when the opponent's behaviors have been observed in the training phase. DRBOA can make agreements in most cases because the opponent modeling is considered in determining the next offer. By contrast, VeNAS(DQN) exhibits a lower agreement rate because the scalability of DQN is not sufficient in large domains.

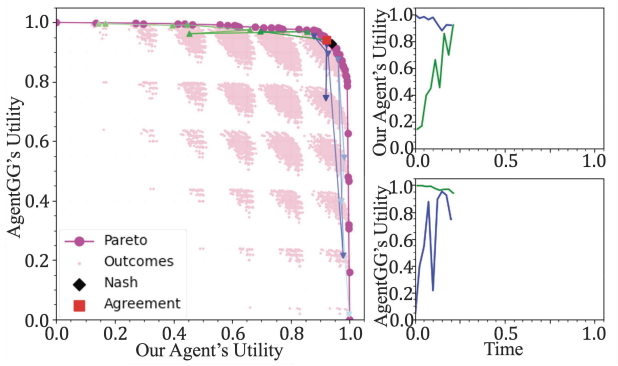
Behavior. We qualitatively analyzed the behavior of MiPN under the policies acquired though learning by observing their common behavior. Figure 3 shows typical examples of MiPN's behavior. MiPN initially made offers with

	Laptop	ItexvsCypress	IS_BT_Acquisition	Grocery	thompson	Car	EnergySmall_A
$\|\Omega\| \propto \|\mathcal{A}\|$	10	10^2	10^2	10^3	10^3	10^4	10^4
Opposition	Low	High	Low	Low	High	Low	High
Heuristics	0.842	0.528	0.843	0.756	0.537	0.741	0.508
DRBOA	**0.883**	0.662	0.869	0.837	0.704	0.822	0.610
VeNAS(DQN)	0.796	0.610	0.805	0.800	0.464	0.510	0.359
PPO-VeNAS	0.845	0.558	0.869	0.763	0.622	0.894	0.562
MiPN	0.848	**0.699**	**0.874**	**0.851**	**0.746**	**0.944**	**0.738**

Table 2: Domain size and average utility score. Our reward base MiPN is more advantageous for regions with larger domain sizes. By averaging the opposing agents, the universal information of the learning method is considered for domain augmentation.



(a) Car-Boulware



(b) Car-AgentGG

Figure 3: Typical examples of MiPN's behavior. In the outcome spaces, the blue lines and circles indicate the agent's behavior and the green lines and triangles indicate the opponent's behavior. In the time-util graph, the blue line represents the change in the utility of the agent and the green line represents the change in the utility of the opponent as time increases. The vertical axis of the upper time-util graph indicates the scores of the agent's utility function, and the vertical axis of the lower time-util graph indicates the scores of the opponent's utility function.

high utility for the opponent, and gradually increased its offers when the opponent made concessions. This behavior is similar to exploring the acceptance threshold of the opponent and functions as an opponent model. The utility function of the opponent can be estimated because it makes offers that have high utility for the opponent but are not accepted. Their behavior can be adjusted considering their opponents' strategies. For example, it sticks to the negotiation until just before the deadline if the opponent's strategy makes a concession just before the deadline (example, *Atlas3* etc.). Conversely, it tries to accept the opponent's offer in the early stage if the opponent's strategy is difficult and will not make any concession (e.g. *AgentGG* and *Hardheaded* etc.). MiPN can predict the opponents' weights of each issue (w_i) correctly. It does not stick to the issue that its own weight is not high and the opponent's weight is high. Thus, the better solutions are found by making a concession effectively toward the issue that is critical for both sides.

Conclusion and Future Work

This study proposed a reward-based negotiating agent strategy utilizing an issue-based represented deep policy network. The reward-based negotiating agent strategy considers the rewards of RL as the generalized signals and does not incorporate the utility function that is rich and concrete information in practice. MDP and multi-issue policy network model including issue-based state and action representations were defined for bilateral negotiations. The scalability of negotiation domains was improved by considering the characteristics of multi-issue negotiation and policy-based learning. The experimental results revealed that the agent with the multi-issue policy network achieved comparable or higher utility than RL agents and champions with heuristics, especially in large-sized domains. Comparative experiments clarified the significant properties of negotiation strategies with RL that the negotiation strategies based on policy network try to make agreements with high step efficiencies.

An elaborative reward function should be considered in the future by considering some characteristics of negotiations. Learning negotiation agents that can work effectively in a different domain and with a different opponent from the training should be studied. Other machine learning techniques such as transfer-learning and meta-learning are required to solve this problem (Sengupta, Nakadai, and Mohammad 2022).

Acknowledgments

This study was supported by JSPS KAKENHI Grant Numbers 22H03641, 19H04216 and JST FOREST (Fusion Oriented REsearch for disruptive Science and Technology) Grant Number JPMJFR216S.

References

Baarslag, T.; Aydoğan, R.; Hindriks, K. V.; Fujita, K.; Ito, T.; and Jonker, C. M. 2015. The Automated Negotiating Agents Competition, 2010-2015. *AI Magazine*, 36(4): 115–118.

Baarslag, T.; Hendrikx, M. J. C.; Hindriks, K. V.; and Jonker, C. M. 2016. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5): 849–898.

Baarslag, T.; Hindriks, K.; Hendrikx, M.; Dirkzwager, A.; and Jonker, C. 2014. Decoupling negotiating agents to explore the space of negotiation strategies. In *Novel Insights in Agent-based Complex Automated Negotiation*, 61–83. Springer.

Bagga, P.; Paoletti, N.; Alrayes, B.; and Stathis, K. 2020. A Deep Reinforcement Learning Approach to Concurrent Bilateral Negotiation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI2020)*, 297–303.

Bagga, P.; Paoletti, N.; Alrayes, B.; and Stathis, K. 2021. ANEGMA: An Automated Negotiation Model for e-Markets. *Autonomous Agents and Multi-Agent Systems*, 35(27).

Bakker, J.; Hammond, A.; Bloembergen, D.; and Baarslag, T. 2019. RLBOA: A Modular Reinforcement Learning Framework for Autonomous Negotiating Agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS2019)*, 260–268.

Delalleau, O.; Peter, M.; Alonso, E.; and Logut, A. 2019. Discrete and Continuous Action Representation for Practical RL in Video Games. *AAAI-20 Workshop on Reinforcement Learning in Games*.

Faratin, P.; Sierra, C.; and Jennings, N. R. 1998. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3): 159 – 182.

Fatima, S.; Kraus, S.; and Wooldridge, M. 2014. *Principles of Automated Negotiation*. Cambridge University Press. ISBN 9780511751691.

Ilany, L.; and Gal, Y. 2016. Algorithm selection in bilateral negotiation. *Autonomous Agents and Multi-Agent Systems*, 30(4): 697–723.

Inotsume, H.; Aggarwal, A.; Higa, R.; and Nakadai, S. 2020. Path Negotiation for Self-interested Multirobot Vehicles in Shared Space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS2020)*, 11587–11594. IEEE.

Jennings, N. R.; Faratin, P.; Lomuscio, A. R.; Parsons, S.; Wooldridge, M. J.; and Sierra, C. 2001. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2): 199–215.

Kawata, R.; and Fujita, K. 2019. Meta-Strategy for Multi-Time Negotiation: A Multi-Armed Bandit Approach. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS2019)*, 2048–2050.

Kraus, S. 2001. *Strategic Negotiation in Multiagent Environments*. Intelligent robots and autonomous agents. MIT Press. ISBN 978-0-262-11264-2.

Lin, R.; Kraus, S.; Baarslag, T.; Tykhonov, D.; Hindriks, K.; and Jonker, C. M. 2014. Genius: An Integrated Environment for Supporting the Design of Generic Automated Negotiators. *Computational Intelligence*, 30(1): 48–70.

Mohammad, Y.; Viqueira, E. A.; Ayerza, N. A.; Greenwald, A.; Nakadai, S.; and Morinaga, S. 2019. Supply Chain Management World — A Benchmark Environment for Situated Negotiations. In *Proceedings of 22nd International Conference on Principles and Practice of Multi-Agent Systems (PRIMA2019)*, 153–169. Springer International Publishing. ISBN 978-3-030-33792-6.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268): 1–8.

Razeghi, Y.; Yavus, C. O. B.; and Aydoğan, R. 2020. Deep reinforcement learning for acceptance strategy in bilateral negotiations. *Turkish Journal of Electrical Engineering & Computer Sciences*, 28: 1824–1840.

Rubinstein, A. 1982. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, 97–109.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Sengupta, A.; Mohammad, Y.; and Nakadai, S. 2021. An Autonomous Negotiating Agent Framework with Reinforcement Learning Based Strategies and Adaptive Strategy Switching Mechanism. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS2021)*, 1163–1172.

Sengupta, A.; Nakadai, S.; and Mohammad, Y. 2022. Transfer Learning Based Adaptive Automated Negotiating Agent Framework. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI2022)*, 468–474.

Silver, D.; Singh, S.; Precup, D.; and Sutton, R. S. 2021. Reward is enough. *Artificial Intelligence*, 299: 103535.

Sutton, R. S.; Mcallester, D.; Singh, S.; and Mansour, Y. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *In Advances in Neural Information Processing Systems 12*, 1057–1063.

Takahashi, T.; Higa, R.; Fujita, K.; and Nakadai, S. 2022. VeNAS: Versatile Negotiating Agent Strategy via Deep Reinforcement Learning. In *Proceedings of 36th AAAI Conference on Artificial Intelligence, AAAI 2022 (Student Abstract)*.

Tavakoli, A.; Pardo, F.; and Kormushev, P. 2018. Action Branching Architectures for Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 4131–4138.