Clustering-Based Approach to Strategy Selection for Meta-Strategy in Automated Negotiation

Hiroyasu Yoshino and Katsuhide Fujita
Department of Electrical Engineering and Computer Science, Institute of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan

Keywords: Automated Negotiation, Negotiation Strategy, Meta-Strategy, Multiagent System, Reinforcement Learning, Clustering.

Abstract: This study aims to develop an automated negotiation meta-strategy and proposes an approach that automatically selects a strategy based on the opponent from a range of available strategies using clustering techniques. The proposed method groups the possible negotiation strategies into clusters and employs deep reinforcement learning to determine an effective bidding strategy for the representative point of each cluster. This strategy is optimized for the average agent within each cluster, consistently outperforming other agents in the same cluster. An analysis of the number of strategy clusters identified using the proposed method indicates that individual utility tends to increase when the number of clusters is limited. Notably, the highest utility is achieved when there are three clusters. In addition, negotiation simulation experiments demonstrate that the proposed approach yields higher individual utility compared to the previous studies.

1 INTRODUCTION

In recent years, consensus building and interest adjustment techniques among multiple agents with different interests have garnered interest in the study of multiagent systems. In the case of agents acting autonomously according to their own preferences, it is difficult to achieve consensus building by centralized agent management when it is necessary to coordinate interests while maintaining confidential information or when a large number of agents are involved. In such cases, there is a growing interest in automated negotiation as a technology that allows agents to reach a consensus while maintaining their autonomy. Supply chain management and drone delivery are attracting attention as examples of real-world applications of automated negotiation (van der Putten et al., 2006) (Ho et al., 2022). One of the advantages of applying automated negotiation to the real world is that it is expected to reduce the cost and time required for negotiation compared to human negotiation. However, there are various problems in applying automated negotiation to the real world, one of which is that, in reality, there are a wide variety of negotiation scenarios and strategies of negotiation opponents. Therefore, there is a need to develop a general-purpose automated negotiation agent that can handle not only specific negotiation strategies and scenarios but also a variety of negotiation strategies and scenarios.

Since it is dificult to deal with various negotiation opponents and scenarios with only a single strategy, a meta-strategy approach has been proposed as a method to develop a general-purpose automated negotiation agent, in which the agent maintains multiple strategies and switches strategies according to negotiation opponents and scenarios (Ilany and Gal, 2016). In this case, it is imperative to consider how to select an appropriate set of multiple strategies and how to employ them effectively in negotiations.

Recently, in the field of automated negotiation, many methods have been proposed to train agents' strategies using reinforcement learning (Razeghi et al., 2020) (Bagga et al., 2021). Moreover, a novel method that integrates this reinforcement learning methodology combined with the aforementioned meta-strategy has been proposed (Sengupta et al., 2021). In this method, the bidding strategy is learned against each of the multiple negotiating agents using reinforcement learning, and the choice of which strategy should be employed is determined based on the history of offers extended by the other agents. However, it is important to note that newly-learned strategies for emerging agents may either be added to or replace the existing set of retained bidding strategies.

Yoshino, H. and Fujita, K.
Clustering-Based Approach to Strategy Selection for Meta-Strategy in Automated Negotiation.
DOI: 10.5220/0012349300003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 256-263
ISBN: 978-989-758-680-4; ISSN: 2184-433X

This can reduce the number of strategies retained and may limit the system's ability to respond to unknown agents and scenarios. Furthermore, since the selection of the bidding strategy for a negotiation is primarily based on the utility obtained by the opponent's offer, the characteristics of the other agent might not be sufficiently captured. Therefore, developing an algorithm that selects strategies according to the opponent characteristics while preserving a diverse array of strategies is crucial.

This study proposes a strategy selection method using clustering in meta-negotiation strategies. The objective is to develop a general-purpose negotiation agent that is capable of effectively responding to various negotiation strategies and scenarios. In addition, we examine and clarify the optimal number of clusters for effective strategy selection by assessing them in terms of individual utility and social welfare.

Our approach in this study involves an algorithm that divides existing negotiating agents into multiple clusters through clustering techniques and switches the effective bidding strategy for the agent positioned at the representative point of each cluster, a strategy learned through deep reinforcement learning. The number of clusters was assessed by conducting simulation experiments with various cluster quantities. We also conducted a comparative experiment with the methodology of a previous study to demonstrate that the proposed strategy selection method can achieve a better utility value. The following is a summary of the contributions of this study.

- Proposal of a strategy selection algorithm using clustering for meta-strategy in automatic negotiation
- Investigation of the number of clusters for strategy selection using clustering
- Confirmation of the effectiveness of the proposed method and its applied negotiation agents for various negotiation strategies and scenarios through simulation experiments

The rest of the paper is organized as follows: Section 2 provides some related works. Section 3 provides a detailed explanation of negotiation settings. In Section 4, we describe the proposed strategy selection algorithm. Section 5 provides the results of the experimental evaluations of negotiation agents applying the proposed method. Finally, we conclude this paper in Section 6.

2 RELATED WORK

2.1 Opponent Features for Automated Negotiation

Opponent features are information consisting of four features derived from the negotiation history (Renting et al., 2020). Renting et al. proposed these features for automatically determining the best configuration of the agent. However, we use these to create groups of strategies with clustering. We provide a detailed description of each feature below.

2.1.1 Concession Rate

Concession Rate is the metric that represents the degree of concession by the opponent agent (Baarslag et al., 2011), denoted by

$$\mathrm{CR}(x_o^-) = \begin{cases} 1 & \text{if } u_o(x_o^-) \leq u_o(\omega^+) \\ \frac{1-u_o(x_o^-)}{1-u_o(\omega^+)} & \text{otherwise} \end{cases} \quad (1)$$

where x_o^- is the bid that gives the minimum utility to the opponent during the negotiation, and $u_o(x_o^-)$ is the minimum utility that the opponent can obtain. Also, ω^+ is the bid that gives the maximum utility to our agent. If $\mathrm{CR} = 1$, then the opponent is fully conceded.

2.1.2 Average Rate

Average Rate expresses the average utility value demanded by the opponent agent as a ratio according to the negotiation scenario. The definition is as follows:

$$\mathrm{AR}(\bar{x}) = \begin{cases} 1 & \text{if } u_o(\bar{x}) \leq u_o(\omega^+) \\ \frac{1-u_o(\bar{x})}{1-u_o(\omega^+)} & \text{otherwise} \end{cases} \quad (2)$$

2.1.3 Default Configuration Performance

Default Configuration Performance is the index that normalizes the utility value obtained by our agent in reaching an agreement according to the negotiation scenario. The DCP value is defined as

$$\mathrm{DCP}(\omega_\mathrm{a}) = \begin{cases} 0 & \text{if } u(\omega_\mathrm{a}) \leq u(\omega^-) \\ \frac{u(\omega_\mathrm{a})-u(\omega^-)}{1-u(\omega^-)} & \text{otherwise} \end{cases} \quad (3)$$

where ω_a is the final agreement bid, and ω^- is the bid that gives the maximum utility to the opponent.

2.2 Autonomous Negotiating Agent Framework with Deep Learning-Based Strategy Selection

Ayan et al. proposed an automated negotiating agent framework that consists of base negotiators, a classi-

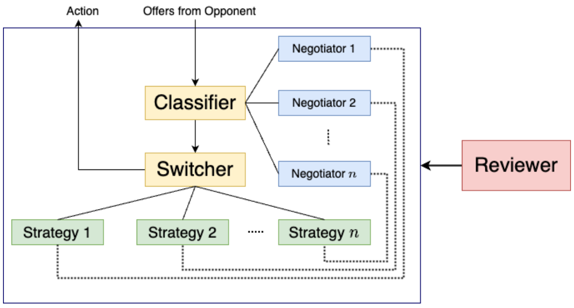

Figure 1: Overview of the negotiating agent framework of the previous study. The classifier (yellow block) predicts the probability that the opponent behaves as each base negotiator (blue and green blocks), and the switcher selects the next action based on the output of the classfier. The reviewer (red block) are used between a negotiation session.

fier, a switcher, and a reviewer (Sengupta et al., 2021). An overview of this framework is shown in Figure 1.

Base negotiator is the pair of negotiating agents and the bidding strategies effective for each. The classifier is based on a one-dimensional convolutional neural network. The input to the classifier is the history of the utility value $U_s(\omega_i)$ that the agent can obtain. The output is the estimated probability that the offer history of the opponent is that of each base negotiating agent.

The switcher is a strategy-switching mechanism based on the classifier for classifying the behavior of an unknown negotiating agent without using an opponent model. The next offer ω_t is selected as follos:

$$\omega_t = U_s^{-1}(u_{s_i}) \text{ where } s_i \in S_b = \{s_1, s_2, \cdots, s_n\} \quad (4)$$

where S_b is the set of base strategies corresponding to the set of base negotiators and u_{s_i} is the utility value based on the strategy s_i

The reviewer is a mechanism that decides whether to add or update new agents and strategies to the set of base negotiators. When a new negotiation agent N_{new} is introduced into the reviewer, it first learns a bidding strategy S_{train} that is effective against it. Then, the agent with the learned S_{train} and the current agent negotiates with N_{new}, and the evaluation value is obtained using the *Eval* function. The evaluation value of strategy s for agent N, $\text{Eval}(N,s)$, is defined as the average utility obtained by the agent using strategy s for agent N. The reviewer gives permission to add a new agent and a new strategy to the set of base negotiators if the evaluation value for N_{new} is higher for S_{train} than for its own current agent, and the classifier is trained on the new set of base negotiators. It then compares the newly trained strategy S_{train} with the base strategies and updates the base strategies with S_{train} and the evaluation value, where the base strategies are the strategies that are effective for each agent in the base negotiator.

3 BILATERAL MULTI-ISSUE NEGOTIATION

Bilateral multi-issue negotiation deals with negotiations conducted by two agents in a common negotiation domain. The negotiation domain defines the issues and the options for each issue in negotiations, which is determined by the set of issues $I = \{I_1, I_2, \cdots, I_n\}$ consisting of n issues, and the set of options V_i consisting of m options for each issue I_i. A bid that the agent proposes during negotiations is a set of options, one from each issue, written as $\omega = \{v_1, v_2, \cdots, v_n\}$.

Each negotiating agent has a preference profile and a utility function. The preference profile comprises the weights of the issues and the evaluations of the options on each issue, and the utility function outputs the utility value of each bid based on the utility information. The preference profile and utility function of each agent are private and not available to other agents. The utility function $U_s(\omega)$ for a bid ω is expressed as

$$U_s(\omega) = \sum_{i=1}^{n} w_i \cdot \frac{\text{eval}(v^i_{k_i})}{\max_j(\text{eval}(v^i_j))} \quad (5)$$

where w_i is the weight of issue I_i and $\text{eval}(v_i)$ is the evaluation value of the option v_i. w_i satisfies $\sum_{i=1}^{n} w_i = 1$ and $w_i \geq 0$, and the evaluation value satisfies $\text{eval}(v^i_j) \geq 0$. The value obtained according to the utility function $U_s(\omega)$ for the agreed-upon bid ω is called the utility value, which is represented by a real number in the range $[0,1]$. The goal of this negotiation problem is to maximize the utility value as much as possible. In this study, we use the Alternating Offers Protocol (Rubinstein, 1982), which is widely used in bilateral negotiations, as the negotiation protocol. In this protocol, both agents take turns performing their actions until the time limit is reached or the negotiation is terminated. When selecting their own actions, the agents choose one of the following three actions.

- Accept: The agent accepts and agrees to the last bid offered by the opponent and terminates the negotiation.
- Offer: The agent rejects the opponent's offer and proposes a new bid to the opponent. Negotiations continue.
- End Negotiation: End the negotiation without reaching an agreement.

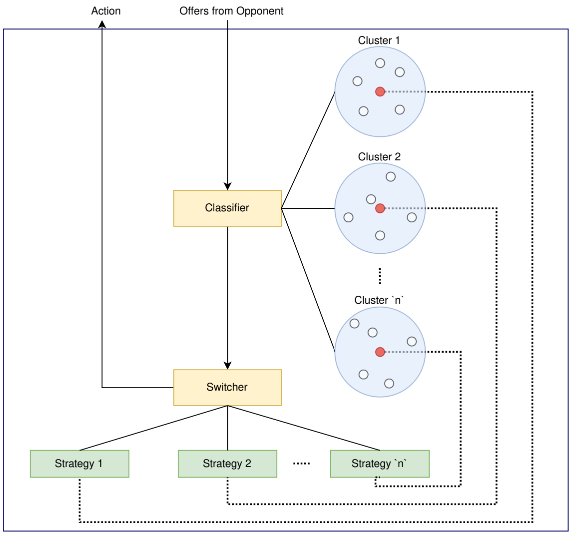

Figure 2: Relationship between each component of the proposed model. The existing negotiating agents are divided into n clusters (blue blocks), and the red point is the representative point of each cluster.

After the negotiation, the agents obtain the utility value for the last bid if an agreement is reached. Otherwise, they obtain the reservation value.

4 STRATEGY SELECTING ALGORITHM BY CLUSTERING

In this section, we propose a new strategy selecting algorithm using clustering for meta-strategy in automated negotiation. Our proposed method consists of three main components: base negotiator, classifier, and switcher, based on the previous study (Sengupta et al., 2021). Figure 2 shows the relationship between each component of the proposed model. As in the previous study (Sengupta et al., 2021), the state space, action space, and reward function for training each effective bidding strategy are defined as

$$s_t = \{t_r, U_s(\omega_s^{t-2}), U_s(\omega_o^{t-2}), U_s(\omega_s^{t-1}), U_s(\omega_o^{t-1}), U_s(\omega_s^{t}), U_s(\omega_o^{t})\} \tag{6}$$

$$a_t = u_s^{t+1} \quad \text{such that} \quad u_r < u_s^{t+1} \leq 1 \tag{7}$$

$$R(s_t, a_t, s_{t+1}) = \begin{cases} U_s(\omega_a) & \text{if there is an agreement } \omega_a \\ -1 & \text{for no agreement and } s_{t+1} \text{ is terminal state} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where t_r is the relative progress time, and U_s is the self-utility function. ω_o^t and ω_s^t represent the proposed and counter offers, respectively. u_r is the self-reservation value, and u_s^{t+1} represents the utility value of the counter offer at the next time step. The classifier calculates the probabilities that the opponent is behaving as each agent of the base negotiator based on the history of the opponent's offer. The switcher determines an action based on the strategy that seems most appropriate from the output probabilities of the classifier.

Our approach improves the classifier model from the previous study and proposes a new method for selection of the base negotiator using clustering.

4.1 Classifier with Opponent Features

The input to the classifier is the history of three features from the opponent features (Renting et al., 2020): concession rate (CR), average rate (AR), and default configuration performance (DCP). Each feature is calculated at every time step, and the input vector to classifier is defined as follows:

$$I_c^t = \{f_o^i\}_{i=t-k}^{i=t-1} \quad \text{where} \quad f_o^t = (\mathrm{CR}_o^t, \mathrm{AR}_o^t, \mathrm{DCP}_o^t),\ k \in \mathbb{Z}_+,\ \text{and } k > 1 \tag{9}$$

where CR_o^t, AR_o^t, and DCP_o^t are the features of the opponent agent at time step t. The opponent's utility function is required to obtain the exact value of each feature, but since the preference profiles are usually kept private, we estimate the opponent's preferences using the opponent model. In addition, to obtain the DCP value, the self-utility value obtained when reaching an agreement is needed. However, as it is impossible to obtain this value while negotiations are in progress, we approximate it by the utility value of the last opponent's offer.

The output of the classifier is the estimated probabilities of behaving as each agent, defined as

$$p_i = \frac{d_i}{\sum_j d_j} \quad \text{where} \quad d_i = \sum_{l=t-k}^{t-1} \frac{f_o^l \cdot f_{N_i}^l}{\|f_o^l\| \|f_{N_i}^l\|} \tag{10}$$

where N_i represents the ith base negotiator.

4.2 Selection of Base Negotiator with Clustering

We create groups of possible negotiation agents and select the pair of representative points of each cluster and an effective bidding strategy corresponding to it as the base negotiator. CR, AR, and DCP are used for clustering along with the classifier. Let us denote the set of negotiation agents as A and the feature vector on agent $a \in A$ at time step t as f_a^t. Then,

$$f_a^t = \frac{1}{|A|} \sum_{b \in A} f_{b \times a}^t \quad \text{where} \\ f_{b \times a}^t = (\mathrm{CR}_{b \times a}^t, \mathrm{AR}_{b \times a}^t, \mathrm{DCP}_{b \times a}^t) \tag{11}$$

where $f_{b \times a}^t$ represents the feature vector on agent a at time step t when negotiating with agent b. In contrast to the classifier model, the utility function of existing negotiation agents is known. Therefore, the opponent model is not used but the actual utility values are used to compute the features. For clustering, the feature vectors at the time step when reaching an agreement are used, and those are calculated while the negotiation is ongoing as they are required in the classifier. In addition, DCP for clustering is calculated with the utility value of the last opponent's offer initially as in the classifier model and with that of the final agreed offer once agreement is reached. Furthermore, we define the feature vector at time step t' after agreement is reached in Equation (12) in order to allow comparison of the feature vectors at any time step, because the number of time steps required to reach an agreement differs depending on the combination of negotiating agents.

$$f_a^{t'} = f_a^{t_a} \qquad {}^{\forall} t' \in \{t_a + 1, t_a + 2, \cdots, T\} \tag{12}$$

where t_a is the time step when agreement is reached, and T is the deadline, maximum time step.

5 EXPERIMENTAL RESULTS

5.1 Negotiation Settings

5.1.1 Negotiation Agents

For training, we used an agent that makes random action choices (Random Negotiator), Boulware agent (Faratin et al., 1998), and Naive tit-for-tat agent (Baarslag et al., 2013) for the method of the previous study. For our proposed method, in addition to the above three agents, we used agents available in Genius platform (Lin et al., 2014), including agents from ANAC 2015-2017 shown in Table 1, because it is desirable to have a certain number of agents for proper clustering.

5.1.2 Negotiation Domain

To compare our proposed method with the one used in the previous study, the same domain should be used for learning of effective bidding strategies. Therefore, we train the strategies in the Lunch domain used in ANAC 2013 in common, in which the reservation value is kept zero and the discount factor for the utility value $\delta = 1$. When the reservation value is zero, the utility value the agent obtains when the negotiation fails is 0, and when the discount factor is 1, the utility value does not decrease over time. The self-utility function is fixed, and the opponent's utility function is randomly generated for training strategies.

For evaluations, we used 12 domains of ANAC 2013 including the Lunch domain, as shown in Table 2. We selected these domains as a diverse set of domains based on two metrics: domain size and opposition.

5.1.3 Opponent Model

Our proposed method estimates the opponent's preference with Smith Frequency Model (van Galen Last, 2012).This method relies on the analysis of the frequency of occurrence of each option within the bids presented by the opponent. In the estimation of weights, issues with frequently appearing options are expected to be important to the opponent; therefore, the weight of the issue is estimated as the ratio of the occurrence frequency of the option with the largest frequency at each issue. The evaluation score for each option is estimated as the ratio of occurrence frequency of each option at each issue.These are denoted as follows:

$$\hat{w}_i = \frac{\max_j(\mathrm{freq}(v_j^i))}{\sum_j \mathrm{freq}(v_j^i)}, \quad \widehat{\mathrm{eval}}(v_j^i) = \frac{\mathrm{freq}(v_j^i)}{\sum_k \mathrm{freq}(v_k^i)} \tag{13}$$

where v_j^i is the jth option of the ith issue, and freq() represents the number of the occurrence frequency of each option.

5.2 Experimental Settings

In the experiments, we examine the number of clusters maintained by an automated negotiating agent using our proposed method, which we call Meta-Strategy with Clustering agent (MSC-agent), and compare its performance with that of a replicated implementation of a prior method, RL-agent.

For training the bidding strategies, we used Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as a reinforcement learning algorithm and OpenAI gym (Brockman et al., 2016) as a training environment. Table 3 shows the hyperparameters for training with SAC. The negotiation simulations were conducted on NegMAS (Mohammad et al., 2021), an automated negotiation platform. In addition, the fuzzy c-means method (Bezdek et al., 1984), one of the soft clustering methods, was used for clustering the agents.

Table 1: Agents available on Genius used for learning.

	Agents
ANAC 2015	AgentNeo, AgentX, AresParty, Atlas3, DrageKnight, JonnyBlack, ParsAgent, RandomDance, SENGOKU, TUDMixedStrategyAgent, AgentBuyooMain, AgentH, CUHKAgent2015, kawaii, MeanBot, PokerFace
ANAC 2016	AgentSmith2016, Caduceus, ClockworkAgent, Farma, GrandmaAgent, MyAgent, Ngent, SYAgent, Terra,
ANAC 2017	AgentF, Farma17, PonPokoAgent, TucAgent
Other	Group2, Group3, Group5, Group6, Group7, Group9, Group10, Group11

Table 2: ANAC 2013 domains used for our experimental evaluations.

Domains	Opposition	Size
Acquisition	0.104	384
Animal	0.15	1152
Coffee	0.279	112
Defensive Charms	0.193	36
Dog Choosing	0.002	270
Fifty-Fifty	0.498	11
House Keeping	0.13	384
Kitchen	0.219	15625
Lunch	0.246	3840
Planes	0.606	27
Ultimatum	0.319	9
Wholesaler	0.128	56700

Table 3: Hyperparameters for Training Bidding Strategy using Soft Actor-Critic Algorithm.

Parameter	Value
Epochs	1000∼5000
Initial collect steps	500
Replay buffer capacity	1000000
Batch size	128
activation fn	relu
Optimizer	Adam
Actor learning rate	3e-5
Actor loss weight	1
Critic learning rate	3e-2
Critic loss weight	0.5
α learning rate	3e-3
α loss weight	1
Initial α	1.0
Target update τ	0.005
Target update period	1
TD error loss function	MSE
γ	0.99

The time limit per negotiation session is 100 rounds, and the agents negotiate only on a single domain when learning the bidding strategy and on all domains in the evaluation. The Self Utility Benchmark and the Utility Against Opponent Benchmark (Sengupta et al., 2021) are used as evaluation measures to compare the benchmark scores of individual utility and social welfare. Each benchmark is denoted as follows:

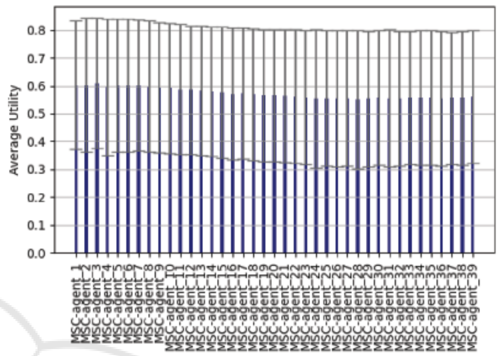

Figure 3: Comparison of benchmark scores for individual utility when the number of clusters n is 1 to 39.

$$S_a = \frac{1}{|A| \times |D|} \sum_{d \in D} \sum_{b \in A} U_a^{a \times b:d} \tag{14}$$

$$O_a = \frac{1}{(|A|-1) \times |D|} \sum_{d \in D} \sum_{b \in A/a} U_b^{a \times b:d} \tag{15}$$

where $U_a^{a \times b:d}$ represents the average utility obtained by agent a against b in domain d over 100 runs. A and D is the set of agents and domains respectively.

5.3 Experimental Results

5.3.1 Comparison of Number of Clusters

We first compared the number of clusters in negotiation simulations, varying the number of clusters maintained by the agents to determine the appropriate number of clusters. Figure 3 shows the comparison of benchmark scores for individual utility when the number of clusters n is 1 to 39. The mean score of individual utility is higher when the number of clusters is smaller. In particular, the number of clusters $n = 3$ has the highest mean individual utility score.

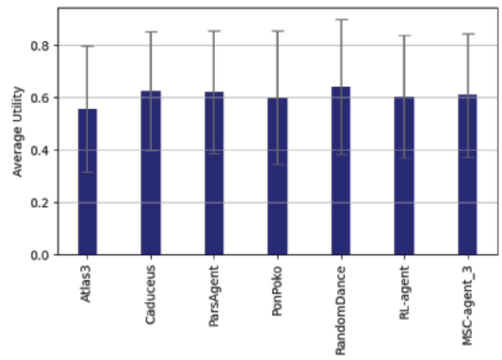

Figure 4: Comparison of MSC-agent with RL-agent and 5 ANAC agents in benchmark scores of individual utility.

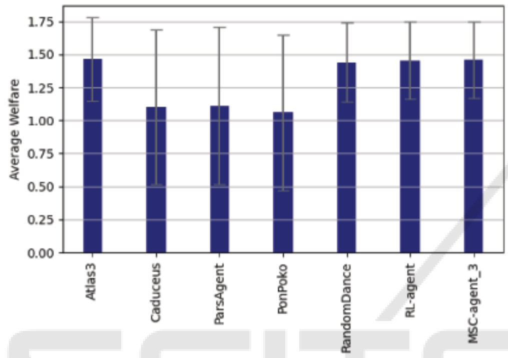

Figure 5: Comparison of MSC-agent with RL-agent and 5 ANAC agents in benchmark scores of social welfare.

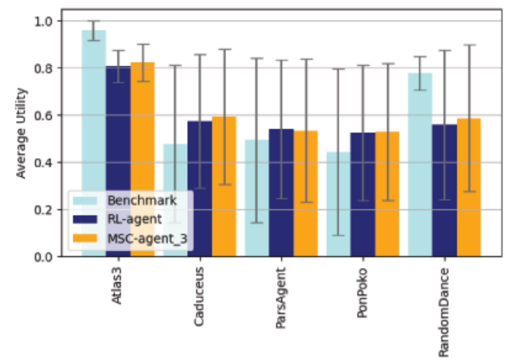

Figure 6: Comparison of Utility Against Opponent Benchmark of individual utility.

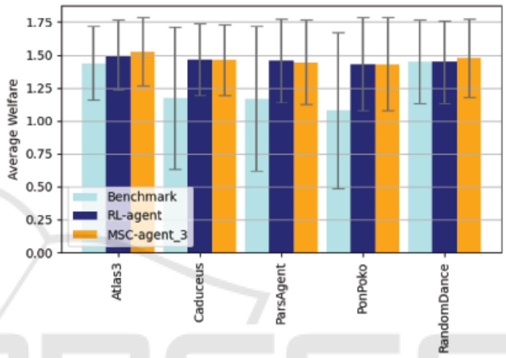

Figure 7: Comparison of Utility Against Opponent Benchmark of social welfare.

5.3.2 Comparison of MSC-Agent with RL-Agent and ANAC Winning Agent

To demonstrate the effectiveness of our proposed method, we compared the best performing MSC-agent (the number of clusters $n = 3$) with the RL-agent, which uses the prior method (Sengupta et al., 2021) and the agents that achieved excellent results in ANAC. The results are shown in Figure 4 and Figure 5. The MSC-agent outperforms the RL-agent in both individual utility and social welfare. In addition, in comparison with the ANAC winning agents, its individual utility score is comparable to that of ParsAgent, Caduceus, and RandomDance, and outperforms the other agents. The standard deviation of social welfare is smaller than that of Atlas3 and RandomDance, which have the same level of utility, indicating that high social welfare is obtained regardless of the domain.

5.3.3 Comparison of Utility Against Opponent Benchmark

We compared the Utility Against Opponent Benchmark for each agent with the individual utility gained by the RL-agent and the MSC-agent for each agent. Figure 6 shows that the MSC-agent achieved higher individual utility for four out of five agents compared to the RL-agent. The MSC-agent also obtained higher individual utility for the three agents compared to the benchmark scores.

Figure 7 shows that the MSC-agent received higher social welfare than the RL-agent against Atlas3 and RandomDance, and compared with the benchmarks, the MSC-agent outperformed all the ANAC winning agents.

6 CONCLUSION

In this study, we proposed a clustering-based strategy selection algorithm for meta-strategy in automated negotiation. For strategy selection, we proposed an algorithm that selects an appropriate strategy based on the opponent features each time during the negotiation with clustering techniques. We trained a bidding strategy effective for each agent positioned at the representative point of each cluster using a deep reinforcement learning algorithm, SAC. Upon analyzing the number of strategy clusters identified by our proposed method, it is evident that the individual utility tends to increase when the number of clusters is limited, with the highest utility achieved when there are three clusters. In addition, negotiation simulation experiments demonstrated that our approach yields higher individual utility than those of previous studies.

Although this study provides valuable insights into meta-strategy for automated negotiation, there remain several avenues for future studies. One potential direction could be the development of new features for agent clustering. In our proposed method, three features from the existing opponent features are used for clustering to group the opponent agents. However, these features focus on the distribution of offers at the end of the negotiation or at specific time points, thereby not sufficiently considering the transition of offers. Therefore, for precise classification of the behavior of the opponent agent, the incorporation of new features that can reflect the negotiation process will be essential.

REFERENCES

Baarslag, T., Hindriks, K., and Jonker, C. (2011). Towards a quantitative concession-based classification method of negotiation strategies. In Kinny, D., Hsu, J. Y.-j., Governatori, G., and Ghose, A. K., editors, *Agents in Principle, Agents in Practice*, pages 143–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

Baarslag, T., Hindriks, K., and Jonker, C. (2013). A tit for tat negotiation strategy for real-time bilateral negotiations. *Studies in Computational Intelligence*, 435:229–233.

Bagga, P., Paoletti, N., Alrayes, B., and Stathis, K. (2021). A deep reinforcement learning approach to concurrent bilateral negotiation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv e-prints arXiv:1606.01540*.

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

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv e-prints arXiv:1801.01290*.

Ho, F., Geraldes, R., Gonçalves, A., Rigault, B., Sportich, B., Kubo, D., Cavazza, M., and Prendinger, H. (2022). Decentralized multi-agent path finding for uav traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):997–1008.

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

Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., and Jonker, C. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence: an international journal*, 30(1):48–70. Online verschenen september 2012; hard copy februari 2014.

Mohammad, Y., Nakadai, S., and Greenwald, A. (2021). Negmas: A platform for situated negotiations. In Aydoğan, R., Ito, T., Moustafa, A., Otsuka, T., and Zhang, M., editors, *Recent Advances in Agent-based Negotiation*, pages 57–75, Singapore. Springer Singapore.

Razeghi, Y., Yavuz, C. O. B., and Aydoğan, R. (2020). Deep reinforcement learning for acceptance strategy in bilateral negotiations. *Turkish J. Electr. Eng. Comput. Sci.*, 28:1824–1840.

Renting, B. M., Hoos, H. H., and Jonker, C. M. (2020). Automated configuration of negotiation strategies. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 1116–1124, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109.

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*, AAMAS '21, page 1163–1172, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

van der Putten, S., Robu, V., La Poutré, H., Jorritsma, A., and Gal, M. (2006). Automating supply chain negotiations using autonomous agents: A case study in transportation logistics. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, page 1506–1513, New York, NY, USA. Association for Computing Machinery.

van Galen Last, N. (2012). Agent smith: Opponent model estimation in bilateral multi-issue negotiation. In Ito, T., Zhang, M., Robu, V., Fatima, S., and Matsuo, T., editors, *New Trends in Agent-Based Complex Automated Negotiations*, pages 167–174, Berlin, Heidelberg. Springer Berlin Heidelberg.