Improving a Reinforcement Learning Negotiating Agent's Performance by Extracting Information from the Opponent's Sequence of Offers

Arpit Agrawal
Supervisor(s): Bram Renting, Pradeep Murukannaiah
EEMCS, Delft University of Technology, The Netherlands

22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

With the prospects of decentralized multi-agent systems becoming more prevalent in daily life, automated negotiation agents have made their place in these collaborative settings. They are an approach to promote communication between the agents in reaching solutions that are better for all involved.

Recent literature has shown great potential in using machine learning, particularly model-free deep reinforcement learning like Proximal Policy Optimization (PPO), to develop more performant automated negotiation strategies. This work focuses on using information from the opponent's sequence of offers in a bilateral negotiation to further improve a baseline PPO agent. This involves extracting and representing information from the opponent's sequence of offers into a state vector with a fixed dimension to modify the input to the agent's policy, and then comparing the utilities this modified agent achieves to the baseline PPO agent. Since there is a large variety of numerical measures to represent a sequence of offers, an ablation study is conducted to investigate the effectiveness of each.

The modified agents consistently reached solutions that had higher social welfare, although the agent's own utility did not improve or diminish significantly in comparison to the base PPO agent.

1 Introduction

Intelligent agents are being embedded rapidly into daily life. As a result, there is a need to study collaborative agents more, with DeepMind's exploration into Hanabi in 2020 being a prime example of such focus [6]. There exist scenarios where multiple agents need to collaborate to reach the most optimal action; examples include search and rescue missions, self-driving cars, and medical decision-making aids. In these scenarios, agents must negotiate with each other to collaboratively agree on a solution and thus the next action. However, much of the current research focuses on competitive games involving resource management and competition and not so much on cooperative settings where actions need to be agreed on by multiple parties.

Although automated negotiation agents have primarily been created with heuristic approaches [1], Bagga et al. [4] and Bakker et al. [5] have recently shown high potential for a machine learning approach to creating agents. Particularly, Bagga et al. [4] demonstrate the effectiveness of their variation of a model-free deep reinforcement learning algorithm in the context of training an automated agent partaking in bilateral negotiations in an e-market that outperforms existing strategies and exhibits adaptive behavior in unknown environments. Similarly, the agent in this paper uses the Proximal Policy Optimization (PPO) [11] model-free deep reinforcement learning algorithm.

The research question this paper aims to answer is: *Can a reinforcement learning negotiation agent's performance be improved with the information from the opponent's sequence of offers?*. An improved agent would consistently reach negotiation agreements that have higher utility for the agent and the opponent in comparison to the current baseline PPO agent.

Sengupta et al. [12] have shown that by analyzing the sequence of offers made by the opponent, the agent's negotiation strategy can be adapted to perform significantly better. This demonstrates the importance of the sequence of offers, but their research uses this information to switch between a set of predefined strategies. As a result, there is potential to instead use the information gained from the opponent's sequence of offers to improve an already effective strategy using PPO that does not need to rely on preexisting heuristic strategies. Ultimately, this insight that the information from the sequence of offers contributes significantly to the effectiveness of an agent's strategy is to be incorporated into the base agent that uses PPO.

Introducing this information into the baseline PPO agent involves representing the opponent's sequence of offers into a state vector with a fixed dimension which would be the input to the agent's policy. As a fixed dimension is needed for the state, while the length of sequence of offers increases as negotiation progresses, numerical measures are employed. However, the numerical measures that can represent a sequence are diverse, and as such, an ablation study is conducted to effectively investigate the contributions of each measure. Section 4 provides further details into this process.

2 Related Work

There are two main categories of related work: work showing the effectiveness of using the opponent's sequence of offers to determine the agent's strategy, and, literature demonstrating the effectiveness of various machine learning approaches used by automated negotiation agents, specifically ones that use reinforcement learning.

2.1 Learning from the Opponent's Offers

Sengupta et al. [12] discuss that the complexity of automated negotiation prevents a single strategy being dominant over all strategies in the variety of negotiation environments and scenarios. As a result, their paper focuses on classifying the opponent's behavior and crucially having a mechanism in the agent that lets it switch between existing strategies to benefit from multiple "experts" within a negotiation session. Importantly, the paper demonstrates that the information from the opponent's sequence of offers is sufficient for an agent to learn to effectively switch and select different, contextually more performant strategies. This suggests that the knowledge of an opponent sequence of offers can have a considerable effect on this paper's PPO agent's strategy. However, Sengupta et al. [12] do not cover using this understanding to develop an agent that does not rely on swapping between existing strategies but rather learns a custom heuristic-less strategy that allows it to adapt to the opponent's sequence of offers.

2.2 Machine Learning Approaches used by Automated Negotiation Agents

Existing automated negotiation agents use a variety of machine learning approaches. Choi et al. [7] design an agent that uses genetic algorithms in an attempt to learn the opponent's preferences using their offers on a stochastic approximation. Zou et al. [15] combines evolutionary algorithms and reinforcement learning by using reinforcement learning to decide when the evolutionary algorithm should evolve, outperforming classic evolutionary algorithms like genetic algorithms. Although proven effective, these algorithms are less feasible for modern negotiation settings since they require a large number of rounds before they have an effective strategy. Yu et al. [14] propose an agent that uses the Bayesian updating rule to update its belief about the opponent's negotiation parameters, which is then used to adapt its concession strategy in bidding to maximize its own utility.

More recently automated agents that use reinforcement learning are being studied. For instance, Bakker et al. [5] propose a modular reinforcement learning based BOA (Bidding strategy, Opponent model, and Acceptance condition) [2] framework that implements an agent that uses Q-learning to learn its bidding strategy. It is important to note, though, that a Q-learning model can suffer from the curse of dimensionality. Crucially, Bagga et al. [4] show that their agent's use of a model-free deep reinforcement learning algorithm performs better than well-known existing strategies, plus the agents that use the algorithm perform well and adapt to different scenarios without needing to be adjusted or reworked.

3 Background

The agent performs in the Stacked Alternating Offers Protocol (SAOP), a negotiation protocol commonly used in automated negotiation research, for example. In brief, agents send bids alternating, and can choose to accept the opponent's bid before sending theirs. The goal in such a negotiation is to maximize the agent's utility (u) expressed by the given preferences profile they have in the negotiation. Such a protocol often has a deadline, in this paper represented by the time available before the negotiation is dropped and the agents resort to their reservation value.

The agent in design has a variation of the component-based approach of Bidding strategy, Opponent model, Acceptance Strategy (BOA) agent architecture as described by Baarslag et al. [2]. Importantly, the agent uses PPO, which results in some additional vital components in the agent's architecture. Table 1 gives a short overview of the main components in the agent. Notably, due to the nature of PPO, the agent does not have a specific Acceptance Strategy component, since the policy that is trained outputs a goal, which if reached by received bid the agent ends the negotiation by accepting the bid, or the agent uses its Bidding Strategy component to make a counter-offer.

The architecture of the agent using PPO is shown in Figure 1. The flow of the primary process involves observing the opponent's most recent bid, updating the state and Opponent Model with the observation, and feeding the state to the policy to receive goal utilities for the agent (u_{goal}) and the opponent's goal utility (u_{opp_goal}). If the goals have not been met by the received bid, the Bidding Strategy then attempts to find the best bid that would most satisfy the outputted goals to send as a counter-offer.

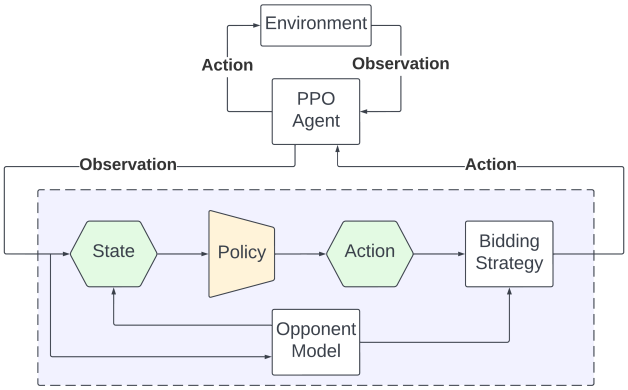


Figure 1: Overview of the Proximal Policy Optimization reinforcement learning negotiation agent's main process.

Importantly, the policy needs to have a fixed number of inputs and outputs, referred to as the dimension of the state space and the dimension of the action space, respectively, in the context of PPO. However, the goal is to learn from the opponent's sequence of offers, the dimension of which increases as the negotiation progresses. Therefore, the focus is on representing the sequence of offers within a fixed dimension of state space which offers an increase in performance in comparison to a base PPO agent that does not learn from the opponent's sequence of offers, represented as an increase in the utility reached by the agent.

4 Methodology

4.1 Implementation

The primary challenge to overcome was representing a sequence of offers as a fixed dimension of state space to be used as an input to the policy, since the sequence of offers would change in dimension as the negotiation progresses. This paper uses numerical statistics to describe the sequence of offers instead, which results in a set of attributes that can represent the list of values instead of inputting the values themselves into the policy [10]. Table 2 gives an overview of the numerical measures considered in this paper and the hypothesised intuition into their effectiveness in improving the agent's policy. As a result of using these measures, regardless of the number of offers received by the agent, the dimension of the input to the policy remains fixed.

The technical implementation for the PPO agent is available on GitHub[1]. Crucially, the state that is inputted into the policy is created in the *select_action* function within the *ppo_agent.py* file. Here the numerical measures as described in Table 2 are implemented and sent to the policy as a state vector. The values u_{goal} and u_{opp_goal} that the policy outputs as its action vector is then used in the agent's bidding strategy, which is described in Section 3.

4.2 Experimentation

Since there are 7 available statistical measures, plus the base agent's progress within a negotiation session and the utili-

[1] https://github.com/brenting/negotiation_PPO

Component	Description
Bidding Strategy	The strategy the agent uses to send their next bid. Often uses the opponent model on top of the strategy to make a bid that the agent estimates to have a good chance of being accepted and has a good utility.
Opponent Model	A model of the opponent that is constructed during negotiation that is used to estimate the opponent's utility of a given bid. The Frequency model strategy is used, shown to be effective by Baarslag et al. [3]
Policy	The component that is updated during Proximal Policy Optimization. The weights on the input and the neural network nodes are updated following the reward function during training, while during runtime the policy outputs actions based on the inputted state.
State	The state is the input to the policy. The state can vary in dimension, but an increase in dimension results in increased complexity for the policy and vice-versa.
Action	The action component represents the output of the policy. In this agent, the output is used by the Bidding Strategy to determine the next action for the agent.

Table 1: Brief overview of a PPO agent's main components.

ties of the past 3 offers from the opponent inputs, an ablation study will be conducted to have a clear understanding of each input's contribution to the performance of the agent. Each version of the agent within this study is trained by pitted against 17 of the 27 available existing agents developed by students in the CSE3210 Collaborative Artificial Intelligence course at the Delft University of Technology by building on the findings in the Automated Negotiation Agents Competition (ANAC) [1].

Similar to this paper's PPO agent, the agents used for training and testing this agent also follow SAOP and are implemented in the GENIUS framework [9]. These agents' domains and preference profiles to be used during negotiation are generated randomly: for the bilateral negotiation 2 profiles are randomly created using the full set of issues and values. Additionally, the number of issues and the size of the bids in the domains are also randomized. To maintain reproducibility, this is all generated pseudo-randomly using a fixed seed. The remaining 10 of the 27 available existing opponents, thus none of the agents in the training set are within the test set, are used in testing the developed agent.

Additionally, each version of the agent is trained 5 times, and results are then generated on the aggregation of the test results of the 5 runs. Since the dimension of the state space can be large for some agent versions, each agent is trained for 6 hours to allow the algorithm sufficient time to train on the more complex inputs. Importantly, each of these agents are trained and tested on the same sets of agents to remove the variability caused by the differing opponents. Negotiation sessions have a deadline of 10 seconds.

5 Results

In this section the results according to the experimental setup in Section 4.2 are presented. Refer to Table 3 for information on each of the agents and how they affect the input state to the policy, specifically columns **Agent** and **Details**.

Figure 2 displays the resulting utilities collected from the ablation study. Due to the nature of the results initially collected (Agents B and S1-S6), more state modifications were tested and are displayed already in order to clarify effects. This is discussed in greater depth in Section 6.

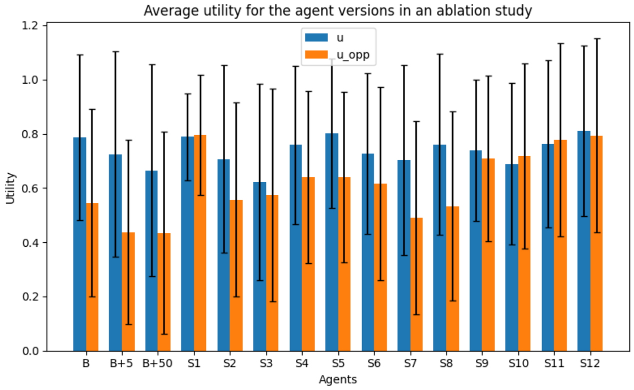


Figure 2: Average utility for the agent versions in an ablation study.

$$u_{social} = u + u_{opp} \tag{1}$$

Social welfare (u_{social}) computed as equation 1 is a simple notion of fairness - the overall utility achieved in the negotiation. During analysis in Section 6 it was noted that the modified agents were reaching negotiation agreements that consistently had higher opponent utilities (u_{opp}), thus Figure 3 displays the u_{social} values for each agent developed.

Table 3 gives an overview of all the results collected during the study, as well as the changes each agent represents on the state.

Finally, a trace of the negotiation session the best performing agent, S12, had with one of the test agents, Agent 78, is compared with the session Agent B had in the same environment in Figure 4.

6 Analysis

Initially, when the 7 numerical measures in Table 2 were used in the ablation study and compared to the utilities achieved by the base Agent B, the results showed insignificant changes in the agent's achieved utilities. The average utilities of Agents S1-S7 were hovering around 0.622-0.802, well within the

Numerical measure	Intuition
Dimension (L)	The length of the sequence of offers made by the opponent can provide the policy insight into the importance and weight on the rest of the numerical measures. Understandably, if the negotiation session has just started the general numerical statistics do not provide the full picture of the opponent and what strategy should be employed by the agent.
Sample Mean (μ)	The mean of the utilities the opponent's bids give to the agent provides a simple negotiation environment-independent understanding of the opponent and the current negotiation session between the agent and the opponent.
Sample Standard Deviation (σ)	Demonstrates the variability in the opponent's strategy. A low σ could suggest a hard-lining opponent, for example. [1]
Sample Median (M)	The median can provide the policy with a more stable understanding of the opponent's position in comparison to μ alone, which can strongly vary if the opponent concedes rapidly, for example.
Sample Mode (Mo)	If an opponent strongly favors a certain bid and sends it multiple times, this would reflect in the sample mode as the same utility would be received by the agent, information the policy can use to concede effectively.
Sample Range (R)	The range can indicate how high and low the utilities are reaching during the negotiation, and indicate the profile of the opponent and the overall potential of the current domain.
Correlation (ρ)	The correlation between the utilities the agent has on the received bids and the estimated utilities the opponent has (from the Opponent Model), can provide vital information on the strategy the opponent is using and the likeliness of a negotiation having both the agent and the opponent get high utilities.

Table 2: Overview of the numerical measures used to represent the opponent's sequence of offers and an intuition into their effectiveness for the agent's policy.

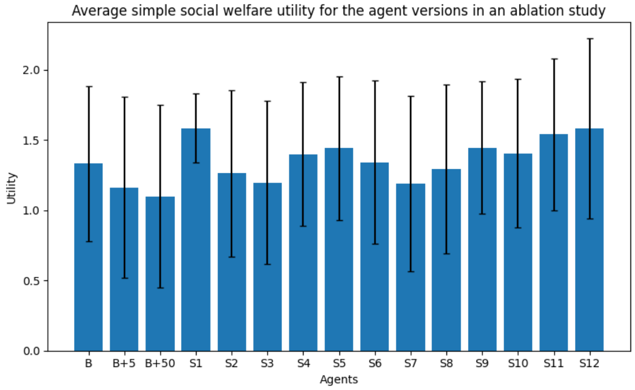


Figure 3: Average simple social welfare utility for the agent versions in an ablation study.

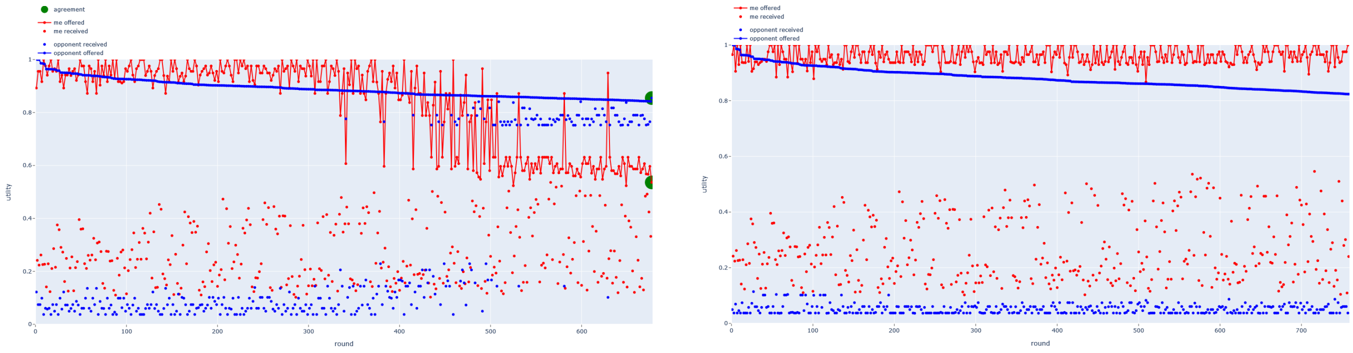
margin of Agent B's 0.779 ± 0.317. However, interestingly, the opponent's utilities have been consistently higher for the modified agents compared to Agent B. As a result, since the utilities of the agents themselves did not change significantly compared to Agent B, but the opponent utilities are higher, the modified agents have a higher u_{social} and therefore are considered more fair.

With this observation, the experiment was extended to add the same numerical measures but instead on the predicted opponent utilities using the Opponent Model component. The goal here was to explore whether this trend of higher opponent utilities without sacrificing the agent's own utility can be continued and further reach higher u_{social} values. Intuitively, the modifications made to Agents S1-S7 are increasing the opponent's utility, therefore adding information about the effects of the opponent's sequence of offers on the opponent's utilities can potentially provide the policy with more resources to work with in maximizing the opponent's utility (while maintaining high own utility). Agents S8-S12 were trained and tested (refer to Table 3 for details) and their results can be seen in Figure 2 alongside Agents S1-S7. Although Agent S7 did not have the most promising results, being the first agent that uses the Opponent Model to predict utilities and compute the ρ between the predictions and the agent's utilities, Agents S8-S12 continue the trend of a mostly stable agent utility, but higher opponent utility.

Figure 3 shows the social welfare computations of each agent. Agents S1-S12 show consistently higher social welfare in comparison to Agent B, and show a mostly inconclusive but slight trend of a gradual increase in social welfare as further numerical measures are added. This trend is more concrete when considering that the results of Agent S1 seem to lie outside the general trend of all the results. Notably, the variance in its achieved utilities between the 5 trained versions of itself are far lower than the other agents. Table 3 shows that its standard error is less than half most of the other agents' standard errors for social welfare. Therefore, it is plausible that since these PPO agents show wide variance in performance each time they are trained, the 5 versions of S1 happened to train 5 stronger than usual agents. A further ex-

Agent	Ablation Level	u	u_{opp}	u_{social}	Dimension	Details
B	-	0.779 ±0.317	0.530 ±0.348	1.309 ±0.567	4	Base (last 3 bids + progress)
B+5	-	0.725 ±0.379	0.437 ±0.341	1.163 ±0.643	6	Last 5 bids + progress
B+50	-	0.665 ±0.390	0.435 ±0.373	1.099 ±0.652	51	Last 50 bids + progress
S1	1	0.771 ±0.178	0.796 ±0.234	1.566 ±0.281	5	B + μ
S2	2	0.707 ±0.346	0.557 ±0.358	1.264 ±0.593	6	S1 + L
S3	3	0.622 ±0.363	0.575 ±0.392	1.196 ±0.582	7	S2 + σ
S4	4	0.759 ±0.291	0.640 ±0.317	1.399 ±0.511	8	S3 + M
S5	5	0.802 ±0.275	0.640 ±0.313	1.442 ±0.513	9	S4 + Mo
S6	6	0.727 ±0.296	0.615 ±0.356	1.343 ±0.578	10	S5 + R
S7	7	0.702 ±0.350	0.490 ±0.357	1.192 ±0.625	11	S6 + ρ
S8	8	0.761 ±0.333	0.533 ±0.348	1.294 ±0.602	12	S7 + μ_{opp}
S9	9	0.739 ±0.260	0.707 ±0.305	1.447 ±0.469	13	S8 + σ_{opp}
S10	10	0.689 ±0.298	0.717 ±0.342	1.406 ±0.527	14	S9 + M_{opp}
S11	11	0.763 ±0.308	0.777 ±0.356	1.540 ±0.539	15	S10 + Mo_{opp}
S12	12	0.811 ±0.315	0.794 ±0.359	1.605 ±0.642	16	S11 + R_{opp}

Table 3: Overview of the ablation study results.



(a) Trace of the utilities in the negotiation between S12 (red) and Agent 78 (blue).

(b) Trace of the utilities in the negotiation between B (red) and Agent 78 (blue).

Figure 4: Comparison of the negotiation sessions Agents B and S12 had with Agent 78.

ploration into these agents and a greater number of versions per ablation level can clear up these doubts.

Overall, Agent S12 averaged the highest performance. It achieved marginally higher utility compared to Agent B, 0.811 ± 0.315 and 0.779 ± 0.317 respectively, while at the same time achieving significantly higher opponent utility 0.794 ± 0.359 compared to 0.530 ± 0.348. As a result, it has the highest social welfare of 1.605 ± 0.642.

However, unfortunately, it is important to note that the results for each of the agents show high variance within their trained versions. As a result, it is quite difficult to make concrete analysis on the contributions of each numerical measure on the performance of the agents in the ablation study, as the differences cannot be directly attributed to the effects of adding or removing the numerical measure from the state. Moreover, even though the number of agents trained and compared is quite large, any specific patterns and trends observed here need to considered critically.

Observing the negotiation traces in Figure 4 shows that the added information from the numerical measures has contributed to Agent S12's strategy in allowing it more acceptable concessions in order to reach an agreement in comparison to Agent B. The negotiation environment and opponent for both of the agents are the same for these two sessions. Agent 78 is a difficult negotiator, acting similarly to the *Hardliner* and the time dependent agents like *Boulware* and *Conceder* described in Baarslag et al. [1]. It gradually concedes as time progresses regardless of its opponent's offers, and accepts once its Acceptance Strategy deems the received utility satisfactory. Although both agents start similarly, by approximately round 350 their strategies diverge. Figure 4b shows that Agent B continues to send hard bids, as it is unable to see any valuable concessions from the opponent, it ends without making an agreement at all. Yet, Figure 4a shows that Agent S12 begins conceding, likely as a result of noticing the very low σ in its utilities from the opponent's sequence of offers, in an attempt to reach an agreement. Thus, Agent S12 is rewarded with a lower but acceptable utility.

This analysis gives insight into how and why Agents S1-S12 achieve higher opponent utilities. The opponent agents were developed in the context of the ANAC [1]. In this context, these agents are given a very low reservation value (the utility an agent receives if an agreement is not made), with the goal of promoting any agreement above disagreements. As a result, since Agents S1-S12 work with the opponent's sequence of offers, they are making concessions and thus reaching more agreements in situations where the baseline Agent B would end up without an agreement. Due to how Agent B is designed, significantly conceding to reach an agreement at low utility is not rewarded and thus reflects the lack of significant change in the agent's u. However, importantly, the opponents are designed to be rewarded for reaching an agreement, which could explain the significant improvement in performance of Agents S1-S12 in social welfare in comparison to Agent B.

To verify that these numerical measures are efficient in providing information to the agent, the base PPO agent was extended by simply including more previous bids (Agents B+5 and B+50), in case the policy is more effective at extracting information from the opponent's sequence of offers than numerical measures. However, as seen in Figures 2 and 3, this led to a decrease in performance. This is particularly evident for B+50, potentially as a result of the massive dimension of the state and resulting difficulty in effectively training the policy for such a large number of features.

7 Responsible Research

7.1 Reproducibility

Several steps have been taken to maintain reproducibility in this research. The implementation of this agent is publicly available on GitHub[2]. As there is no data needed since the domains are pseudo-randomly generated, the agent can be simply trained and tested by running the *train.py* and *test.py* files, respectively. The pseudo-randomness maintains reproducibility in the experiment, and allows one to modify the experiment and train the agents on the same randomizer seed to effectively explore the effects of their modifications with minimal concern of the effects of the random processes in the agent.

As a result, rerunning the experiment as described in Section 4 should result in similar results as presented in this paper, with only variations due to the nature of PPO [11]. There is, however, one dependency of the results that is difficult to control; the algorithm is computationally expensive and time-taking, and therefore improvements or variations in the agent's performance might rely on the hardware it was trained on. This has larger effects if the algorithm is run for a relatively shorter amount of time, which would not allow the policy to settle on less powerful hardware and potentially leading to performance disruptions for the agent. The experiment allowed each PPO agent to train for 6 hours in this paper with the goal of minimizing the potential effects of the hardware. To ensure further reproducibility in the results it is recommended to run this algorithm on a controlled hardware cluster.

7.2 Ethics

The goal of this research is to improve the performance of automated agents in a collaborative environment, where agents work together to reach better solutions than those they would have reached on their own. A few examples of cooperative settings that require agents to agree upon a solution include search and rescue missions, self-driving cars, and medical decision-making aids. Taking an action solely individually without effectively communicating and negotiating with other agents could result in never finding the person is distress due to a lack of consensus on delegating search areas, car accidents, or an inaccurate medical diagnosis. However, the agents developed during this paper focused on increasing the agent's own utility, often leading to a reduction in the opponent's utility, sometimes disproportionately. As a result, agents developed using this method are not sufficient for all situations of collaboration. In cases where a utilitarian solution is more appropriate - for instance it is unimportant who finds the person in distress during a search and rescue mission, the primary mission is to get them rescued as soon as possible - the agents would need to be developed with a focus on social welfare. One one hand, the agents presented in this paper are not yet satisfactory for all situations of cooperation, since they prioritize maximizing their own utility above maximizing the total utilitarian utility between the agent and the opponent. On the other hand, the improved agents did significantly improve the opponent's utility while not negatively affecting the agent's own utility, which meant that it reached negotiation solutions that had better social welfare than the baseline PPO agent.

Although automated negotiation agents have been showing potential [8], negotiation has primarily been a human activity [13]. Therefore, if automated negotiating agents progress into extremely effective negotiators, securing each negotiation regardless of setting to their terms, and the developments are not distributed well, it could lead to the threat of a consolidation of negotiation power. Discrepancies between negotiating performance is of course also human, but one with a supreme automated negotiating agent would have the reach and performance that could be untouchable. Although this can aid humans, this can also be unfair to people also involved in the negotiation but have no ability to have the same level of resources to effectively negotiate their own terms against the superior automated negotiating agent. This goes against the premise of negotiating to collaborate onto a solution, and shifts instead of abusing the collaborative system to maximize personal control. In conclusion, further explorations into automated negotiation agents should keep in mind the potential pitfalls of such an agent.

8 Conclusions and Future Work

The purpose of this paper was to improve a reinforcement learning negotiation agent's, particularly a Proximal Policy Optimization (PPO) agent, performance by extracting information from the opponent's sequence of offers. To do so, an ablation study was conducted to investigate the contributions of the variety of numerical measures that can represent a sequence on the performance of an agent. Importantly, numerical measures were used to represent the opponent's sequence of offers since the state vector the agent's policy takes as input needs to have a fixed dimension throughout negotiation,

[2] https://github.com/brenting/negotiation_PPO

which is not possible for sequence of offers that increase in length every round. The effectiveness of the agents in the ablation study was analyzed and compared to the baseline PPO agent.

Although the ablation study results showed large variance in the performances of each agent trained within an ablation level, the general trend shows that adding the numerical measures in the state vector does not significantly impact the agent's own utilities, but it does positively affect the opponent's utilities. Once further numerical measures were added representing the opponent's predicted utilities for their sequence of offers, it was seen that social welfare continued increasing.

This led to Agent S12 - an agent that had all: 5 numerical measures on its own utilities, 5 measures on the predicted opponent utilities, and the correlation (ρ) and dimension (L) measure between them from the opponent's sequence of offers - having the best performance, particularly when considering social welfare. It was shown that the reason for a lack of significant improvement in the agent's own utilities, but large improvements in the opponent utilities, has to do with how the modified agents, like Agent S12, are more willing to work with the opponent and concede. This caused an increase in the opponent's utilities since the opponents were developed in the context of always preferring an agreement over a disagreement, where disagreements would take place more commonly with the baseline Agent B.

Future work involves more sophisticated PPO agents, where training the same agent again does not result in large variances. From this, several approaches can be taken to further the research this paper has conducted. An ablation study can be conducted again, but this time with focus on the effects and contributions of each and every numerical measure. Such a paper has the potential of finding strong links between effective automated negotiation strategies and the information needed, in this case a numerical measure, to develop it. Furthermore, numerical measures, though effective at representing the opponent's sequence of offers, are not the only way; future work could focus on using a machine learning algorithm to learn the most important or valuable features from the opponent's sequence of offers and use those features instead in the state vector for the agent's policy. Finally, although a baseline agent was chosen which uses PPO, this investigation can be done agents which could be using a multitude of model-free deep reinforcement learning algorithms. There is potentially greater room for improvement in using information from the opponent's sequence of offers in other algorithms of the sort.

References

[1] Tim Baarslag, Katsuhide Fujita, Enrico Gerding, Koen Hindriks, Takayuki Ito, Nicholas Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73–103, 05 2013.

[2] Tim Baarslag, Koen Hindriks, Mark Hendrikx, Alex Dirkzwager, and Catholijn Jonker. *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies*, volume 535, pages 61–83. 01 2014.

[3] Tim Baarslag, Koen Hindriks, Mark Hendrikx, and Catholijn Jonker. Predicting the performance of opponent models in automated negotiation. volume 2, 11 2013.

[4] Pallavi Bagga, Nicola Paoletti, Bedour Alrayes, and Kostas Stathis. A deep reinforcement learning approach to concurrent bilateral negotiation. *CoRR*, abs/2001.11785, 2020.

[5] Jasper Bakker, Aron Hammond, Daan Bloembergen, and Tim Baarslag. Rlboa: A modular reinforcement learning framework for autonomous negotiating agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 260–268, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems.

[6] Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for AI research. *CoRR*, abs/1902.00506, 2019.

[7] Samuel P.M. Choi, Jiming Liu, and Sheung-Ping Chan. A genetic agent-based negotiation system. *Computer Networks*, 37(2):195–204, 2001. Electronic Business Systems.

[8] Michele J Gelfand et al. Negotiating in a brave new world: Challenges and opportunities for the field of negotiation science. *The Psychology of Negotiations in the 21st Century Workplace*, pages 479–500, 2012.

[9] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30:48–70, 02 2014.

[10] DG Rees. Summarizing data by numerical measures. In *Essential Statistics*, pages 24–38. Springer, 1989.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[12] Ayan Sengupta, Yasser Mohammad, and Shinji Nakadai. An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism. *CoRR*, abs/2102.03588, 2021.

[13] G Richard Shell. *Bargaining for advantage: Negotiation strategies for reasonable people*. Penguin, 2006.

[14] Chao Yu, Fenghui Ren, and Minjie Zhang. *An Adaptive Bilateral Negotiation Model Based on Bayesian Learning*, pages 75–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[15] Yi Zou, Wenjie Zhan, and Yuan Shao. Evolution with reinforcement learning in negotiation. *PLOS ONE*, 9:1–7, 07 2014.