Group7 CS 451 Project Report

Talha Özer, Emin Şahin Mektepli, İpek Motorcu

April 2024

1 Introduction

This report details the design and implementation of our negotiation agent, which utilizes a strategy divided into three time-dependent phases. Each phase is implemented to adjust the agent's behavior in response to the upcoming negotiation context, aiming to optimize outcomes and enhance its bargaining leverage. By using our newly implemented techniques and learning from both historical data and real-time opponent behavior, our agent is designed to achieve the best outcome in complex negotiation scenarios. Performance metrics including Outcome Maximization, Adaptability to Partner Behavior, and Agreement Rate are employed to assess the agent's success. Outcome Maximization evaluates the overall utility or value achieved through negotiated agreements and aims to surpass the utility obtained by other agents. Adaptability to Partner Behavior measures the agent's ability to adjust its negotiation approach in response to the behavior of the other party. In phase three, our agent acts according to opponents' behavior, aiming to adapt and secure favorable outcomes. Agreement Rate serves as a primary measure of success, indicating the agent's effectiveness in reaching mutually beneficial agreements. In phase three, our agent aims to reach an agreement by offering bids with lower utility for itself, thus demonstrating its commitment to achieving successful negotiations.

2 NEGOTIATION STRATEGY

This section presents an explanation of the behavioral strategy of the agent during the negotiation. The agent goes through three distinct, time-dependent phases which determine its action patterns.

2.1 Acceptance Strategy

Our agents acceptance strategy is designed to complement it's three-phase bidding strategy. The phase transitions are time-dependent; therefore, the agent conducts different behaviors with time.

Phase 1: Strategic Patience - Initially, our agent sets high acceptance criteria, favoring offers which provide significantly higher utility and are worse for the opponent. This is reflected in our code where the agent is likely to accept an offer only if it substantially exceeds our reserved value while ensuring the opponent's utility is lower, exemplifying a strong opening stance. An experimental coefficient is used to determine how high the offer's utility must be than our reserved value, and the precious insight about the utility function of the opponent is harvested in order to better pose a stern negotiator at the beginning.

Phase 2: Gradual Concession - As the negotiation progresses, the agent's acceptance criteria adapt and loosen. It starts conceding in the sense that it lowers the bar of how high the utility of an offer must be for it to accept. It should be noted that the other criterion that states the utility of the offer for our agent must be greater than that for the opponent persists. This ensures an ever stern agent throughout this phase as well.

Phase 3: Time-Sensitive Concessions - In the final stretch, our agent becomes more agreeable as the time constraint imposes itself heavier. Leaving past conditions behind, it starts to accept any bid that provides a higher utility than the following bid by itself would provide. This is executed with a simple yet effective *'isnextbidworse'* function, which calculates the utility of the bid our agent is about to make and compares it to the last bid made.

Throughout these phases, our agent dynamically adjusts its criteria based on the phase of negotiation and the behavior observed from the opponent. By aligning our acceptance strategy with the phased bidding approach, the agent demonstrates adaptability and strategic foresight, aiming to secure advantageous agreements while navigating the complexities of negotiation dynamics. When we use only ACNext as an acceptance strategy our results are shown in Figures 1 and 2. Results are similar to our original ACNext strategy.

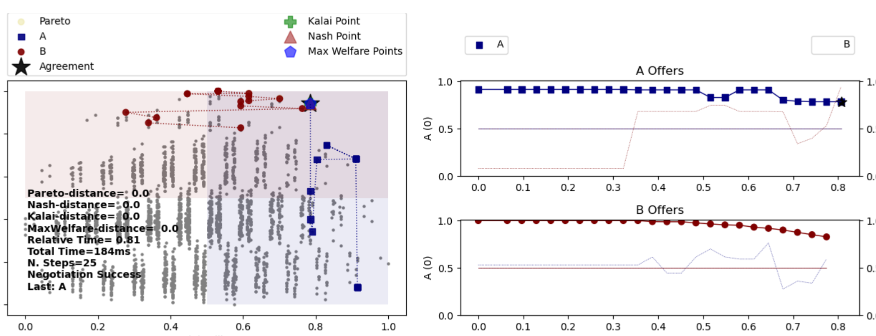


Figure 1: When the opponent is Boulware and ACNext Acceptance Strategy

2.2 Bidding Strategy

Similar to the work of Dirkzwager et al. in [2], we have implemented a bidding strategy with three phases. The transition between these phases depends solely on time. After conducting various tests, we settled on time limits of 0 to 0.33

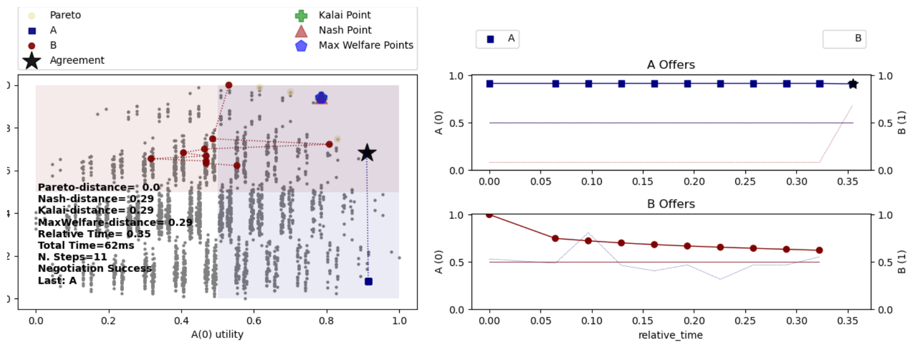


Figure 2: When the opponent is Conceder and ACNext Acceptance Strategy

for phase 1, 0.33 to 0.66 for phase 2, and phase 3 starting from 0.66 onwards.

Initially, the agent is in the first phase, where there is no pressure of time yet. It starts by offering a bid where its utility is as high as possible, and the opponent's utility is as low as possible. In this phase, the agent insists on the same bid even if the opponent is conceding. We chose this one as an opening bid because there is no pressure on time yet and if the opponent is conceding depending solely on time it gives us an advantage.

In the second phase, we utilize the *'concedingbid'* function, where the agent begins to concede from its utility as well. Consequently, it commences offering bids with reduced utility for itself while gradually enhancing the utility of the opponent. Despite our utility decreasing, it remains greater than that of our opponents. We conducted tests to determine the concession rate and have decided on using linear time for phase 2. We can also use opponent behavior to determine the concession rate; however, we want to give the opponent an opportunity to decrease their utility as well. In other words, in the first two phases, the behavior of the opponent is only taken into account to guess its reservation value, unlike the third phase.

The third phase is when the time constraint really starts to impose itself, so both parties are expected to be more concessive. We also use relative time in phase 3 while calculating the best rational value for that time. The agent examines the opponent's behavior with the *"isOpponentConceding"* function, which checks if the opponent has made an offer that is worse for them than their previous offer or if their offer's utility is considerably below the average of their last couple of offers. If they do not cooperate and insist, our agent offers bids where our utility is more than the opponent's and decreases our utility slowly (We utilize the *"concedingbid"* function and provide False as a parameter, indicating that the opponent is not conceding). It means it sometimes proposes offers that increase the utility for itself relative to its previous bid. By doing this, we are trying to increase our utility in agreement. In our original model, if our opponent is not conceding, we only give offers which have maximum utility for our agent. However, this behavior will result in a deadlock; that's why we changed our original model to make our agent concede but slowly. If the opponent starts conceding, the agent begins to offer bids with the opponent's utility higher than its own utility (We utilize the *"concedingbid"* function and provide True as a parameter). This implies a considerable difference in the acceptance strategy.

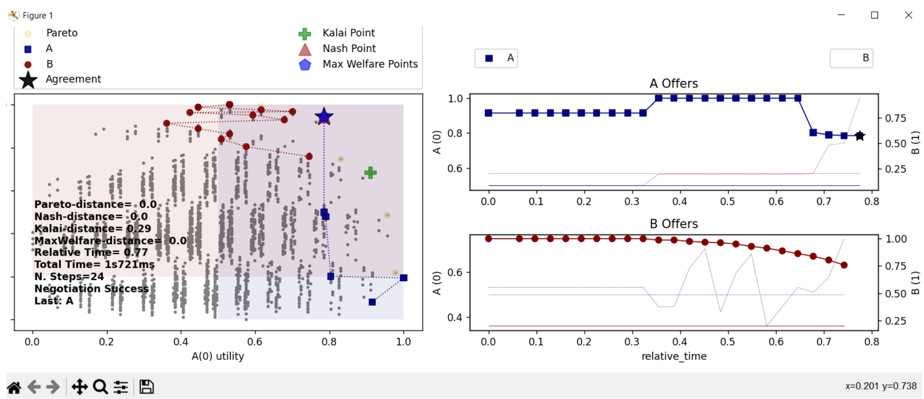


Figure 3: When the opponent is Boulware

Fig.3 shows the results of a negotiation against Boulware. Our agent is represented as A, while Boulware is represented as B. We can easily observe the different behaviors of our agent in different phases. In Phase 1, it insists on the same bid. In Phase 2, it attempts to find a bid where our agent's utility exceeds that of the opponent's and the opponent's utility exceeds their estimated reservation value. If such a bid cannot be found, our agent begins to offer bids that are most favorable to our side. In the third phase, our agent assesses the opponent's behavior and adjusts its bids accordingly. When Boulware starts to concede, our agent also concedes. In this example, agreement is on the Pareto, Nash, and Max Welfare points.

In the earlier version of our model, we only considered relative time without a coefficient. When we used only relative time to calculate the concession rate, the results in Fig. 4 were worse compared to our latest version shown in Fig.5. In the latest version, we divided relative time by a constant coefficient, which improved our results. We determined this constant after conducting several tests.

2.3 Reservation Value Modelling

We have adopted two different approaches in Reservation Value Modelling, which means there are two versions of our model.

The simpler version function as the following: First, the agent assumes the reservation value of the opponent to be the same as its own. Following that it tweaks the assumption with regards to the bids by the opponent. If their current bid provides them with higher utility than our agent's current assumption

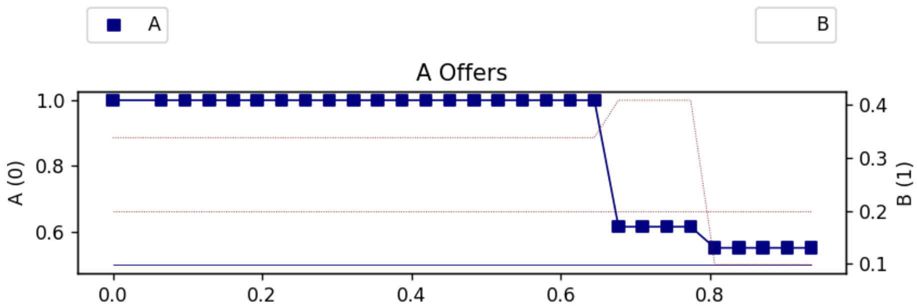


Figure 4: Old Version

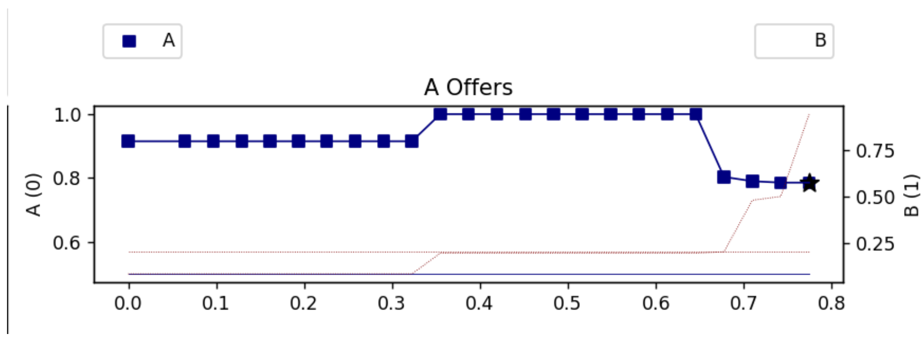


Figure 5: New Version

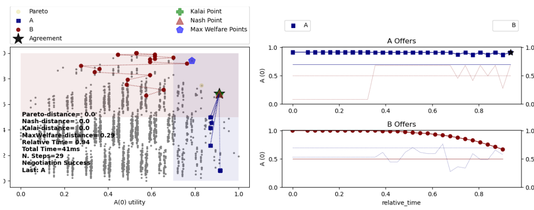
about their reservation value, the agent gets more confident and increases its assumption ever so slightly. However, if the opponent is making an offer which has lower utility for them than the agent's current assumption about their reserved value, clearly the assumption is flawed and it's heavily decreased.

Other version directly borrows the curve fitting principles from existing resources. It keeps track of the bids of the opponent like the previous version, and than uses an analytic approach, trying to find a curve that explains the point on the graph of these bids. We have implemented this method in the agent as well in order to be able to compare different cases, yet our tests did not always suggest a certain and definitive improvement. This could be explained with the fact that the points on the opponent utility graph do not come to be as a result of a single function that could be simply explained but generally is the product of a more intricate decision taking process, which in certain scenarios may need to be analysed with deeper complexity.

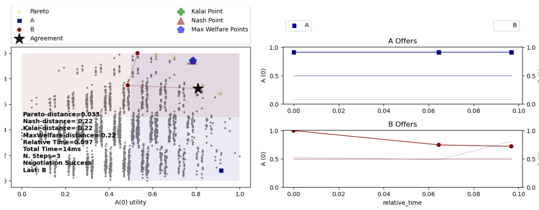
3 PERFORMANCE

3.1 Basic test on Party domain

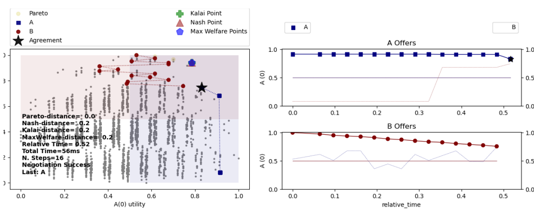
Figure 6: Basic test on Party domain



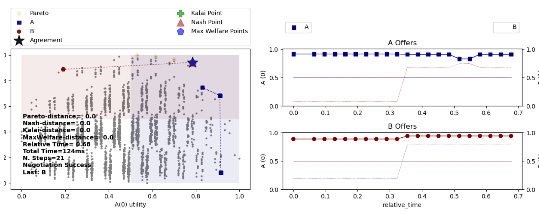
(a) Against Boulware



(b) Against Conceder



(c) Against Linear



(d) Against itself

As shown in Figure 6 our agent achieved pareto optimal in several cases. Reason being our bidding and acceptance strategies.

3.2 Test on other domains

Figure 7 shows our agents can negotiate successfully regardless of the domain. Several negotiations ended on Nash point and on Pareto Frontier.

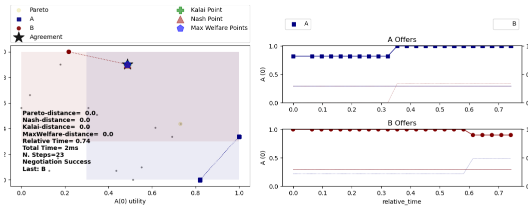
3.3 Test your opponent model

Our reservation value affected our negotiation with Boulware positively, but there wasn't any noticeable difference with Conceder, as shown in Figure 8.

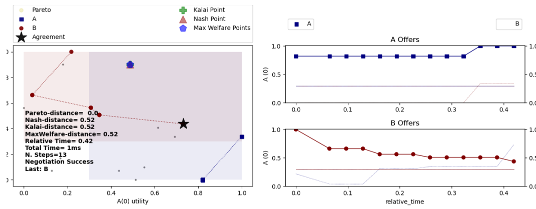
4 Future Perspectives and Conclusion

Analysing the opponent and divining their reservation value is of extreme importance in many scenarios and the present model can be improved with the adaptation of an algorithm that is specifically developed for this occasion. Simple analytic methods do not always succeed at coming up with deeper insights about the opponent, so a study which considers the existing bidding strategies as well may result in a more appropriate solution.

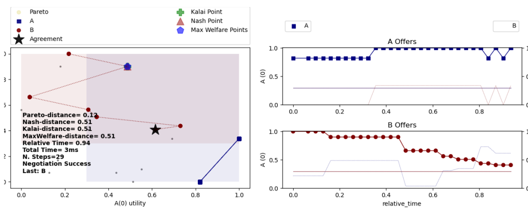
Figure 7: Test on other domains



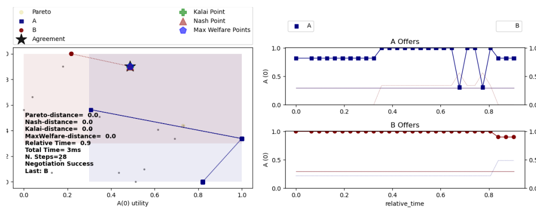
(a) Against Boulware Different Domain



(b) Against Conceder Different Domain

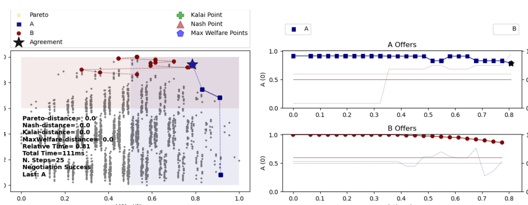


(c) Against Linear Different Domain

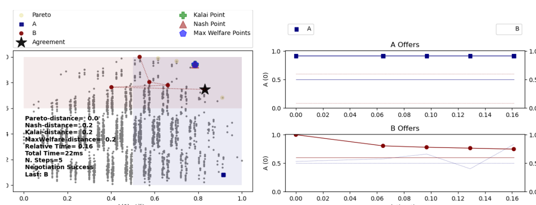


(d) Against itself Different Domain

Figure 8: Test your opponent model



(a) Without Reservation Value Model and Against Boulware



(b) Without Reservation Value Model and Against Conceder

The concession rate which is used to determine how much the agent is willing to concede at a given time is calculated with a simple linear function in our model. This may not be the most appropriate approach because it is only natural to think that a different curve could give better performance since it yields to a different time management strategy. Therefore, how to model and adapt to the time constraint is a question open to further investigation.

To further improve its practicality and effectiveness, the agent could be equipped with adaptive learning algorithms that will allow it to learn from each interaction, thereby continuously improving its negotiation strategies based on past outcomes and evolving scenarios. This self-improving mechanism would enable the agent to handle a broader range of negotiation types and complexities, potentially allowing it to support or even take over negotiations typically performed by humans.

Exploring these extensions will not only enhance the functionality and applicability of our negotiation agent but also ensure that it can perform effectively in diverse and unpredictable real-world negotiation scenarios, achieving optimal outcomes while maintaining fairness and integrity.

5 References

C. R. Williams, V. Robu, E. H. Gerding, and N. R. Jennings, "Iamhaggler2011: A gaussian process regression based negotiation agent," in Complex Automated Negotiations: Theories, Models, and Software Competitions, ser. Studies in Computational Intelligence, T. Ito, M. Zhang, V. Robu, and T. Matsuo, Eds. Springer Berlin Heidelberg, 2013, vol. 435, pp. 209–212. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30737-9 14

A. Dirkzwager, M. Hendrikx, and J. Ruiter, "The Negotiator: A dynamic strategy for bilateral negotiations with time-based discounts," in Complex Automated Negotiations: Theories, Models, and Software Competitions, ser. Studies in Computational Intelligence, T. Ito, M. Zhang, V. Robu, and T. Matsuo, Eds. Springer Berlin Heidelberg, 2013, vol. 435, pp. 217–221. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30737-9 16