Wheelin' and Dealin': Automated Negotiation Agents

By: Nick Case, Michael Erickson, and Hannah Estes

What problem are we addressing?

Working with autonomous agents allows for a controlled environment to isolate the effects of social utility depending on altruistic and selfish agents. The Harvard Law School defines negotiation as “back-and-forth communication designed to reach an agreement” [3]. While this seems trivial, the circumstances of different people, or agents, add an additional layer of complexity. During the negotiation, the commonalities and differences of the two parties will affect the terms of the agreement. The back-and-forth nature allows the two parties to adapt and adjust their initial terms while still keeping their values as a top priority. Finding an acceptable solution provides a benefit to both parties. In some cases, if the differences are too extreme, an agreement might not be reached, which is detrimental. The art of negotiation is commonly referenced as a human-only process due to the complexities of balancing everyone’s values and goals, but there is research taking these concepts and applying them to computer systems.

The term “automated negotiation agents” encapsulates this research of creating computer systems capable of negotiating. With this in mind, an automated negotiation agent can interact with another one of its kind or a human to find a solution that benefits both parties. Amongst this research topic, “Automated Negotiation: Prospects, Methods and Challenges” defines autonomous agents as computer systems that can problem solve, can receive inputs from an environment, have specific goals to achieve, are autonomous, and can be both reactive and proactive [1]. This research proposes that a computer agent will negotiate for a specific person by maintaining the same values and goals as their human counterpart. Autonomous agents can prioritize creating the best possible social utility for their outcome or creating the best utility for themselves when interacting and negotiating with other agents. In short, the agent can be inherently selfish or selfless.

The problem this paper addresses is that autonomous agents are often created with the purpose of being able to negotiate autonomously, but often do not reach the optimal outcome, as defined by achieving Pareto optimal and Nash solution for that situation [4]. Specifically, determining how an altruistic or selfish bias affects the social utility of the society of other automated peer negotiating agents. Working with these autonomous agents allows for a controlled environment to isolate the effects of social utility depending on altruistic and selfish agents.

Why is this problem important?

The processes and concepts behind negotiations are complex. This problem is significant because it showcases that altruistic agents positively impact society. In contrast, selfish agents ultimately do not alter the social utility by influencing other agents to act more altruistically, eventually creating a more selfless environment around them.

How will we address this problem?

We explored the Automated Negotiation League (ANL) competition, a worldwide competition where teams develop automated negotiation agents. This year, the competition has two separate goals: maximize the average utility of your agent relative to that of others and maximize the sum of all utilities of negotiations that the agent is involved in. While the utility of the agent works towards both goals, the first goal requires an agent focused on winning and perhaps undermining their opponent, while the second goal focuses on cooperation and compromise.

Our group has created two unique negotiation agents to compete within the boundaries of this competition. These 16 agents do not come from past competition but instead come from the class CSE3210 at Inha University, dedicated to studying and creating automated negotiating agents. The agents we selected from that pool of CSE3210 agents are agents 2, 3, 14, 18, 22, 24, 245, 27, 29, 32, 33, 41, 61, 64, 67, and 68.

Our agent looking to maximize its individual utility employs a strategy that attempts to negotiate into a solution that favors it over its opponent. Given a list of unique Pareto optimal solutions, there will exist one such that all other Pareto solutions result in a loss for the agent and a gain for its opponent. This solution is likely to bring the maximum possible gain for the cooperative agent. This agent is likely to be "ruder" with others, meaning that it will be willing to "walk away" given an opposing agent that is unwilling to let it have a net gain.
Our agent looking to maximize the sum of utilities of all agents in negotiations will employ a strategy of finding compromise with its peers. This agent will want to avoid the other agent "walking away" or the negotiation timing out as much as possible. This option will result in the minimum possible gain for the cooperative agent, as neither the agent nor its opponent has any utility. The cooperative agent will aim to find the Pareto optimal solution for itself and its opponent that maximizes the sum of both agent's utilities. This solution will serve as the maximum possible gain for the cooperative agent.

Above are the ideas and theories behind creating our selfish and selfless agents, but one key factor is the implementation of these agents. Certain implementations or methods for negotiating lean more towards selfishness or selflessness. For our paper, however, we created our method of negotiating and evaluating bids using the idea of a Pareto Curve, accepting bids outside our estimated curve which slowly decreases in radius over time. This enables our agents to compromise more based on the amount of elapsed time during a negotiation session.

An important aspect of creating this curve is recognizing that in order to find the utility of a bid for an opponent, the agent must build a model for the opponent's proposed bids. To ensure that a reliable model is built, the agent must first consider multiple opponent bids before it starts implementing an estimate curve. At the beginning of a negotiation session, both of our agents only propose very high-value bids to them. These are found by evaluating 5000 possible bids each time it is the agent's turn, scoring them based on the metric the agent is looking to maximize, and proposing the bid with the highest score. The bid with the highest score is remembered from turn-to-turn, meaning that this frequently will converge to a single bid that is repeated each turn. The agent will also reject all opponent bids at this time. Once a certain amount of negotiation time has passed, the agent will begin estimating a pareto curve with its trained opponent model.

With a trained opponent model, all bids can be placed on a x-y plane in which x is the utility for the agent and y is the estimated utility of the opponent. These points will all fall inside $0 \leq x \leq 1$ and $0 \leq y \leq 1$. The agent will use its estimation curve to deny all bids that fall inside the curve and accept any bids that fall on or outside the curve.

Both agents employ the same basic formulae to create their semicircular estimation curves that slowly decrease in size over time. In order to have the radii decrease in size over time, the formula for calculating the radius is $r = \sqrt{1 + b^2} * k^{\frac{t-t_0}{1-t_0}}$ where b changes the point in which the circle initially intersects with y=1 or x=1, k is the rate of the decay of the radius, and t_0 is the percent duration of the negotiation session in which the agent should start evaluating bids coming in based on the model presented. The formulae to accept any bid outside this curve are as follows:

$y > \sqrt{r^2 - (x - x_0)^2} + y_0$ or $x > r + x_0$, where (x_0, y_0) is the location of the center of the curve. If either of these statements are true, then the incoming bid is accepted.

Our agent looking to maximize the sum of utilities of all agents, named WolfpackAgent, implements a curve centered at the origin. This allows the agent to initially only accept bids in which the combined utilities of the agents are very high, then slowly accepting bids that are more in favor of one agent or another.

Our agent looking to maximize its personal utility, named LoneWolfAgent, implements a curve centered in the second quadrant. Being that the curve is semi-circular, this will allow for the acceptance of any bids in which the utility is high for our agent, while being picky about the amount of utility offered to an opposing agent in situations where the utility for our agent is not the absolute best.

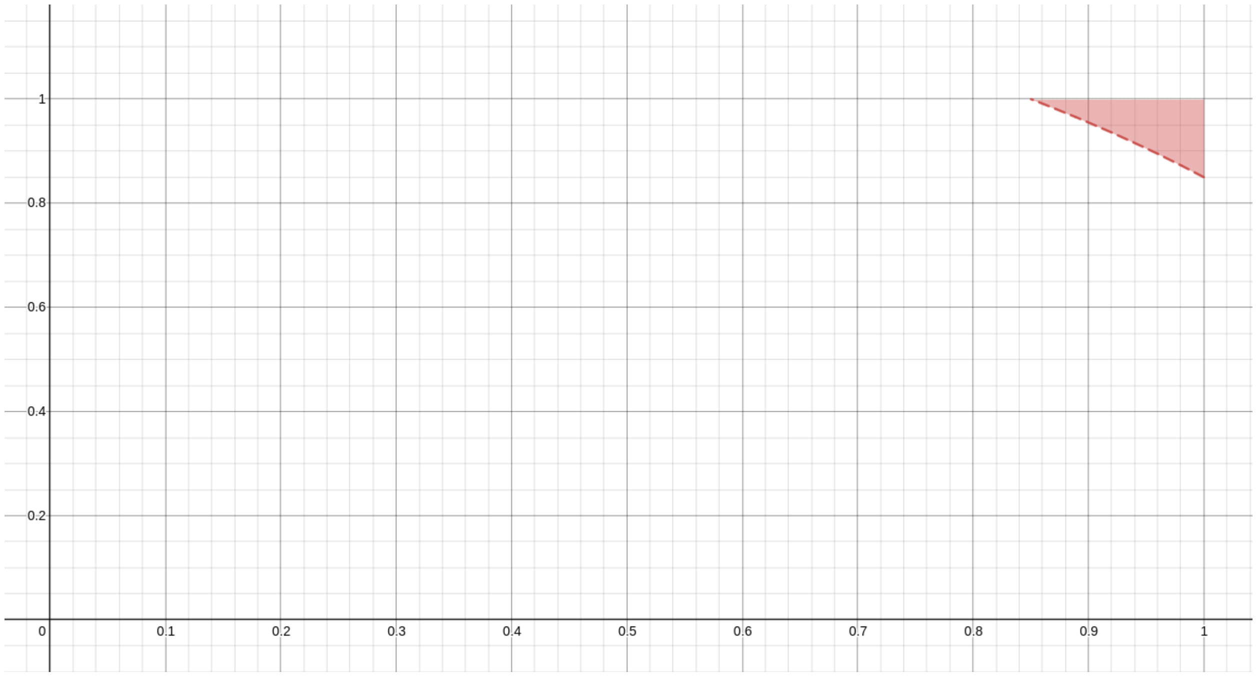

Fig 1: The area shaded red is the domain of bids in which the WolfpackAgent will accept at $t = t_0$. The specific parameters for the function are $t_0 = 0.65$, $b = 0.85$, $k = 0.75$, and $(x_0, y_0) = (0, 0)$.

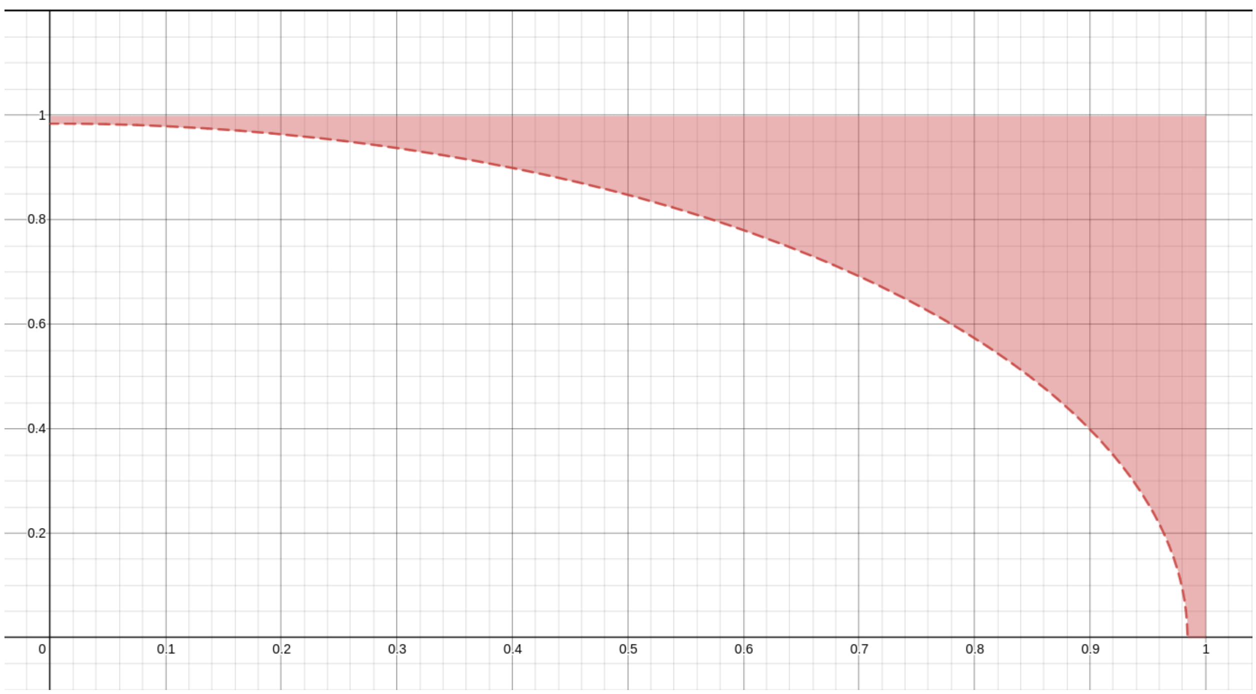

Fig 2: The area shaded red is the domain of bids in which the WolfpackAgent will accept at $t = 1$, or the expiration time for negotiation.

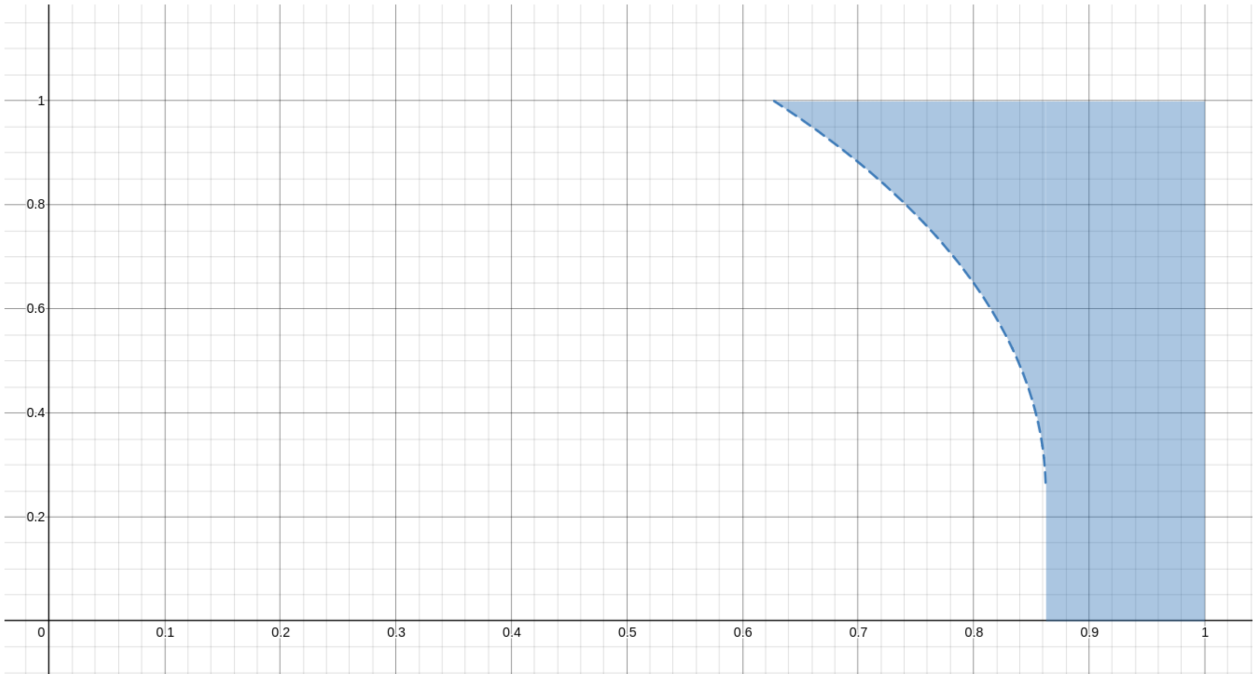

Fig 3: The area shaded blue is the domain of bids in which the LoneWolfAgent will accept at $t = t_0$. The specific parameters for the function are $t_0 = 0.75$, $b = 0.85$, $k = 0.65$, and $(x_0, y_0) = (-0.45, 0.25)$.

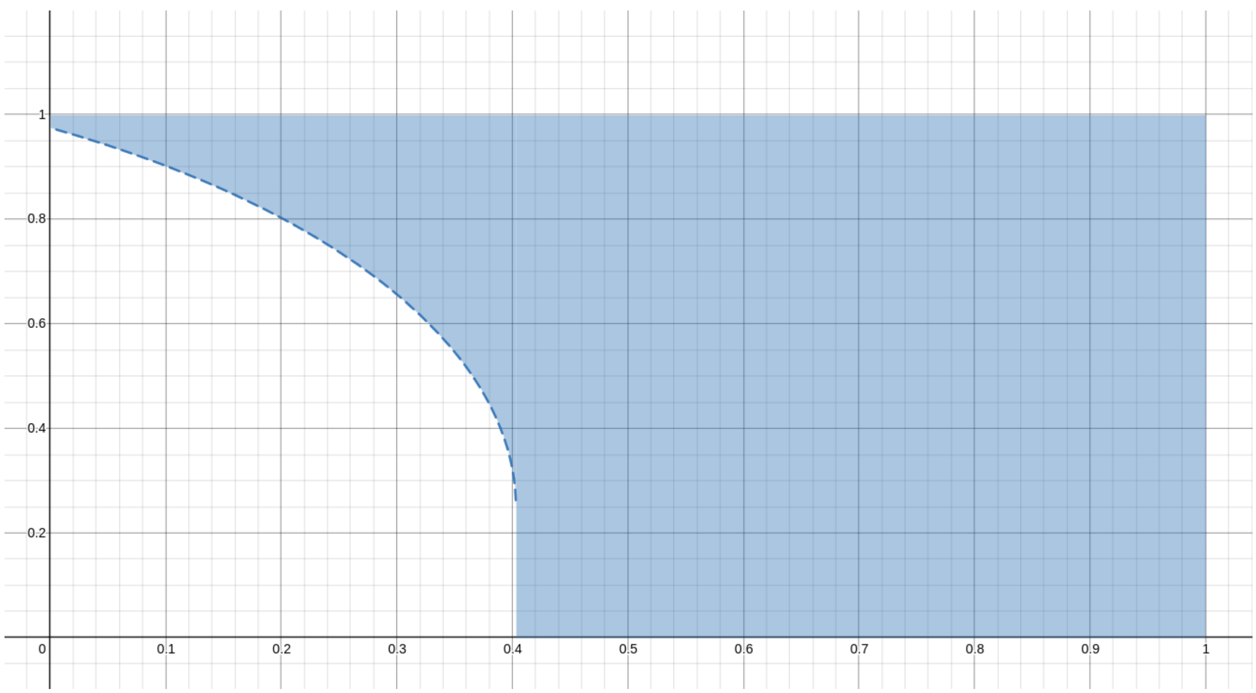

Fig 4: The area shaded blue is the domain of bids in which the LoneWolfAgent will accept at $t = 1$, or the expiration time for negotiation.

The tournaments we ran follow the competition protocol laid out by the league organizers, using the Python GeniusWeb Library and the SAOP, Stacked Alternating Offers Protocol [5][6]. In each negotiation round, each agent will be assigned a unique and random preference profile and will be paired with every other agent. Each preference profile will contain the domain in which the agents are negotiating, and their available bids based on their preference within the domain. To ensure that there is no bias in which agent goes first in the negotiating process, all agents are paired with one another twice, giving each the opportunity to be first. We used two different bid domains, meaning that each agent will run against each other agent four times, twice per domain. This translates to a single tournament involving 480-544 times ($2 * \frac{n!}{(n-2)!}$ times, where n is the number of agents). The domains selected for our use during this paper were domains 26 and 43 within the ANL competition repository. According to SAOP, Negotiations are time-limited and each agent takes turns in negotiations, choosing one of 3 options: make an offer, accept the current offer, or end the negotiation [5]. When an offer is accepted, the utility of the agent is evaluated based upon their preference profile for the current negotiation [4].

For this paper, we ran a total of three types of tournaments, repeating each five times for consistency purposes. The two agents we created, named Wolfpack (social utility focused) and LoneWolf (personal utility focused), competed in two out of our three tournaments. The first type of tournament was just the randomly selected 16 of the CSE3210 agents. The purpose of this tournament was to create a baseline for the societal utility values in order to compare against the impact that our selfish and selfless agent had within the group. The second tournament type was the same random 16 CSE3210 agents, and our Wolfpack agent, allowing us to see the impact of an agent prioritizing social utility within the competition. And our final tournament type was the same random 16 CSE3210 agents from the other runs, but this time, we inserted our LoneWolf agent in there, allowing us to see how average social utility changes when there is an agent that prioritizes its own utility always.

Finally, with the data from these tournament runs, fifteen in total, we compared the runs of the control tournaments versus the data from the tournaments where our agents participated. We evaluated how our two unique agents rank for social utility achieved from the tournament and how the average utility of the other agents changed or were affected by our two agents.

What are some alternatives and how do you justify your approach?

Our group decided to choose to enter agents into this particular competition because of the recency of the competition. There have now been 14 ANL competitions, but choosing the most recent one will allow us to compete most fairly. The 2023 competition uses Python, meaning that our group will be able to use a great deal of Python libraries in competing. The older competitors used Java, so using Python and additional libraries could provide a distinct advantage for our two agents. Alternatively, the older competitions that used Python only had access to their current versions of libraries. By choosing the most recent competition, the libraries that we had access to were the closest to the 2023 competitions. The old competitions might not have been able to access specific libraries or had to overcome certain bugs that we did not have to encounter.

With our implementation, we decided to create an environment that had variety in implementation schemes, but was generally neutral on the selfish to altruistic spectrum. Our justification for this stemmed from the ultimate goal of mimicking a real world scenario. In most cases, people are relatively neutral but might slightly favor altruistic or selfish qualities. The 16 random agents within the CSE3210 class were selected and confirmed they did not have any extreme qualities, yet had varying implementations when it came to negotiation. Alternatively, a more systematic approach could have occurred, in which there were a certain number of neutral, altruistic, and selfish agents. Due to the tournament's thorough approach, an increased variability could provide an alternative outcome or confirm our findings. This is an area for future research.

As another option, rather than taking robots from the CSE3210 class, our environment can consist of the other provided agents such as the hardliner or Boulware agent. These agents provided an additional variable when it came to selecting our environment and seemed to be more focused on winning the competition which allows the robots to learn and maximize social and personal utility.

One of the biggest challenges for this team stemmed from the multitude of possibilities when choosing how to implement our two robots, Wolfpack and LoneWolf. For this research topic, developing unique agents to isolate specific features while controlling the variability of the agents became key. The Wolfpack agent is designed entirely for altruism whereas the Lonewolf is completely designed with the desire of selfishness. As mentioned before, many of the agents provided in previous competitions and classes tried to maximize social and personal utility during different scenarios. Our agents prioritized one ideology above everything else. Additionally, creating the agents from scratch allowed an easier comparison between the two agents and did not have completely different implementations. By taking out the factor of implementation effectiveness, each agent was measured on an even scale where neither agent had an advantage.

How did you evaluate your approach?

As mentioned briefly above, we created fifteen tournaments to create a controlled environment and test each one of our agents. For each environment that the tournament was run, the sixteen CSE3210 agents–2, 3, 14, 18, 22, 24, 245, 27, 29, 32, 33, 41, 61, 64, 67, and 68. Three styles of tournaments were run. First, the control tournament included only the chosen CSE3210 agents. Second, the Wolfpack agent was tested with the sixteen CSE3210 agents. Lastly, the Lonewolf agent was run alongside the sixteen CSE3210 agents. Each tournament style, respectively named Control, WolfpackAgent, and LoneWolfAgent was run five times. After the fifteen tournaments were run, our script calculated and produced output for the average utility of all the agents, the average social welfare of all the agents, and the standard deviation for both average utility and average social welfare.

For each tournament run, information about the negotiations are calculated and stored as output. For each tournament type, all the results were combined and information was averaged for each agent within the tournament. The tournament's data was aggregated and the mean was found for each type. For analyzing data, this was a necessity to find the average across the five different trials. From the averages computed, the average utility of all the agents within the tournament, the standard deviation of average utility, the average social welfare for all agents, and the standard deviation for average social welfare was calculated. After all tournaments completed this process, the results are outputted to an Excel spreadsheet. This spreadsheet is a representation of all the aggregation and averages of all the collected data within the fifteen tournaments.

	Control Mean	Control std	LoneWolfAgent Mean	LoneWolf Agent std	WolfpackAgent Mean	WolfpackAgent std
avg_utility	0.64	0.061	0.64	0.057	0.634	0.057
avg_nash_product	0.408	0.054	0.414	0.055	0.409	0.056
avg_social_welfare	1.28	0.09	1.281	0.101	1.267	0.1
avg_num_offers	810.909	372.095	742.526	379.648	717.364	367.671
count	60	0	64	0	64	0
agreement	55.55	3.766	58.729	4.397	58.141	4.15
failed	4.45	3.766	5.271	4.397	5.859	4.15
ERROR	0	0	0	0	0	0

Fig 5: Output file representing the aggregating and averages of the collected data

Our emphasis on average social welfare and average utility of the agents within each environment encapsulates our goal of the research. By looking at the differences, it provides context on how effective our Wolfpack and LoneWolf agents are in influencing the general mass by increasing social good or disrupting altruism with selfish behavior.

What are your main findings of the project?

From the simulations we have run and the data we have collected, we have concluded that with an agent designed for prioritizing social utility and an agent designed for prioritizing personal utility, there is no significant difference between the average social utility of the entire society with those agents involved, disproving our initial hypothesis.

There are a couple of important thoughts to consider that are suggested from our simulation. Societally speaking, the fact that one agent wants to help those around them or the fact that one agent prioritizes itself does not contribute to the overall societal utility. Drawing parallels between our agents and society, if there is a negotiation over something like a new bus stop or a new law, one person trying to do the best for society or one person doing the best for themselves doesn’t matter to the general utility of society, as long as there is a big enough sample size. One person’s agenda does not matter when there is a large sociometric that is computed from a very large sample size. Perhaps this will change if a certain percentage of people start prioritizing social utility or personal utility, a potential area for future research.

The other important thought that can be taken from our simulation is within the field of automated negotiation agents. When using automated negotiation agents to negotiate, one person or company that creates an agent that tries to get the best of itself or tries to get the best for all cannot affect the end resolution of the negotiation over time. For example, with sponsored search auctions, if one company creates an automated negotiation agent to never compromise and force the search engine to compromise on the selling price, over time, that does not affect the price or value of those keywords in the long run.

Works Cited

[1] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, M.J. Wooldridge, and C. Sierra. 2001. Automated Negotiation: Prospects, Methods and Challenges. *Group Decision and Negotiation* 10, 2 (2001), 199–215. DOI:https://doi.org/10.1023/a:1008746126376

[2] Jonathan Levin. 2009. Sponsored Search Auctions. Retrieved from https://web.stanford.edu/~jdlevin/Econ%20285/Sponsored%20Search%20Auctions.pdf

[3] Katie Shonk. 2022. What is Negotiation? *PON - Program on Negotiation at Harvard Law School*. Retrieved from https://www.pon.harvard.edu/daily/negotiation-skills-daily/what-is-negotiation/

[4] Automated Negotiation League (ANL) 2023. Retrieved from https://web.tuat.ac.jp/~katfuji/ANAC2023/cfp/Automated_Negotiation_League_2023.pdf

[5] Anon. GeniusWebPython. Retrieved November 23, 2023 from https://tracinsy.ewi.tudelft.nl/pubtrac/GeniusWebPython

[6] Reyhan Aydoğan, David Festen, Koen V. Hindriks, and Catholijn M. Jonker. 2017. Alternating offers protocols for multilateral negotiation. *Modern Approaches to Agent-based Complex Automated Negotiation* (2017), 153–167. DOI:http://dx.doi.org/10.1007/978-3-319-51563-2_10