KB time diff Agent Strategy

Kobayashi Yuji

May 1, 2023

Tokyo University of Agriculture and Technology
y-kobayashi@katfuji.lab.tuat.ac.jp

1 Introduction

The Automated Negotiating Agent Competition (ANAC) is an international tournament that has been running since 2010 to bring together researchers from the negotiation community.
I took part in the Automated Negotiation League (ANL) and proposed a ”KB time diff agent”. The agent aimes to get high social welfare.

2 Design

My goal is to get high social welfare. To achieve this, I developed a bidding strategy and an acceptance strategy. However, I did not consider opponent model and learning this time. The main feature of those strategies is that I used time difference to reach an agreement with any agent in time. They are described in more detail in the following sections.

2.1 Bidding Strategy

I describe about my Bidding Strategy. To create it, I made three main attempts.

1. making accept function
2. using sorted bids
3. choosing better bid for getting social welfare

2.1.1 Making Accept Function

I make ”accept function”. This function is based on a sigmoid function. The sigmoid function is shown in Figure1. The reason for using the sigmoid function as a basis is that I wanted to use the characteristic of a gradual change at the beginning and at the end and a sudden change in the middle in my strategy. I transformed the sigmoid function into the following formula for use in this strategy.

$$y = \left(-\frac{1}{1+e^{-10(t-0.5)}}+1\right) \times (\text{highthreshold} - \text{lowthreshold}) + \text{lowthreshold} \tag{1}$$

”t” represents the degree of progress and ranges from 0 to 1. This function can be expressed graphically as shown in Figure2. However, highthreshold is 0.9 and lowthreshold is 0.7 in Figure2.
I would use this function as a threshold for individual utility when deciding a bid.

2.1.2 Using Sorted Bids

My agent sorts all the bids of utilityu and suggests the bids whose individual utility is above the threshold in order. Then, if the threshold value is exceeded, it suggests the bids with the highest individual utility in order again. By doing this, it can propose a bid that has a high utility value for it to the opponent agent without omission. This makes it easier to find a bid that both agents agree on and reduces randomness.

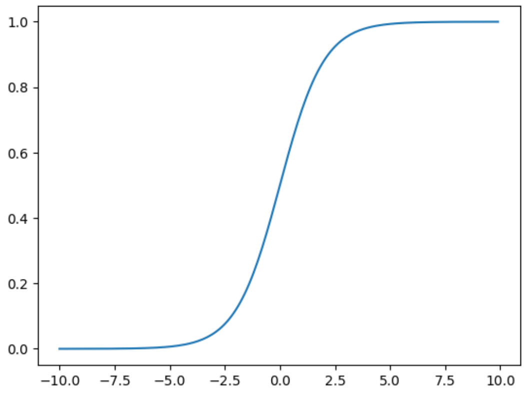


Figure 1: Sigmoid Function

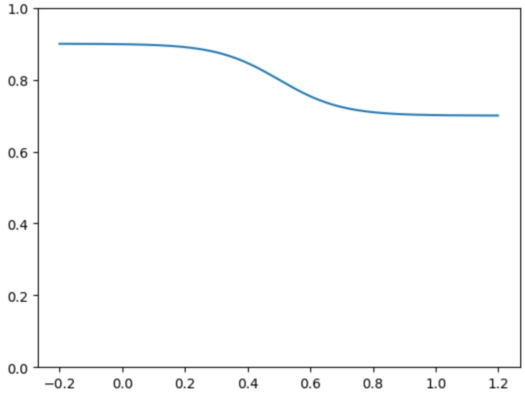


Figure 2: Accept Function

2.1.3 Choosing Better Bid For Getting Social Welfare

When the progress is less than 0.9, the proposal is made in the manner described in the above section, ”Using Sorted Bids”. However, in this approach, it is likely that no agreement will be made, as my agent is only concerned with my own utility value. My goal is to get high social welfare, and the key to this is to make sure that agreement is made, even if my utility value is low. In order to ensure agreement, my agent uses the bid proposed by the opponent agent. First, my agent memorizes the bids proposed by the opponent agent until the progress reaches 0.9. The bid proposed by the opponent agent is expected to be somewhat high. This is because there is usually no significant advantage in proposing a low bid. Next, my agent sorts memorized bids. Then, when the progress becomes 0.9 or higher, the proposal is made in the sorted descending order. Since the proposal should be pleasant for the opponent agent in many cases, I consider that there is a high possibility that agreement will be made. Since my agent also has some high utility value, I expect to obtain a high social surplus as a result.
However, there is a problem. It is a case of a wide time interval when the opponent angent makes a proposal. If the time interval to propose by the opponent agent is wide, the threshold value 0.9 becomes meaningless. This is because there will be a possibility that my agent will not have time to make a proposal when the progress is above 0.9. Therefore, my agent measures the time interval between each time an agent makes a suggestion and receives a response from the opponent agent. Using this time interval to predict the remaining number of times, my agent can make suggestions in the above sort order regardless of progress 0.9. In this way, agreement can be reached in most cases.

2.2 Acceptance Strategy

I describe about my Acceptance Strategy. To create it, I made two main attempts.

1. using accept function

2. setting first threshold and second threshold

2.2.1 Using Accept Function

The same accept function described in the previous section is used here. If the individual utility is greater than the utility obtained by the accept function, the proposal is accepted.

2.2.2 Setting First Threshold And Second Threshold

My agent sets two thresholds for progress. The first threshold is for relaxing conditions. The second threshold is to ensure agreement at the end of the process.

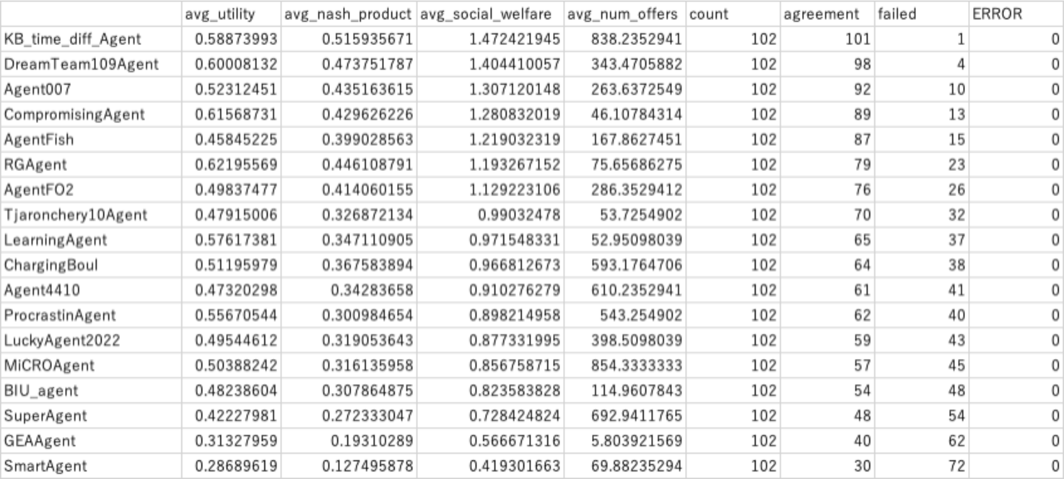
First, I explain the second threshold. The second threshold is the total minus the expected time difference. If progress is greater than the second threshold, my agent always make them agree, regardless of the value of my individual utility.

Second, I explain the first threshold. Holds the bids proposed by the opponent agent up to the first threshold. After the first threshold, if what the opponent agent proposed has already been proposed and your utility value is 0.3 or more, my agent agrees. Holds the bid proposed by the opponent agent up to the first threshold. After the first threshold, if what the opponent agent proposed has already been proposed and your utility value is 0.3 or more, my agent agrees. This is because I believe that the proposals made by the opponent agent so far are favorable to the opponent agent. If the opponent agent has a high utility value and my agent have a certain utility value, then social welfare will increase.

However, there is a problem when setting the first threshold. The basic first threshold is 0.93. As the time interval increases, the second threshold may become smaller than the first threshold. If this happens, the first threshold value should be even smaller than the second threshold value. In this way, the longer time interval can be handled. As a result, agreement is reached in most cases.

3 Result

To test my agent, I tried to fight with almost the agents who competed in ANL 2022 using three domains. However, the time limit was set to 1000ms. The results are shown in Figure3. As a result, my agent won first place in social welfare. And it was agreed in allmost negotiations.



	avg_utility	avg_nash_product	avg_social_welfare	avg_num_offers	count	agreement	failed	ERROR
KB_time_diff_Agent	0.58873993	0.515935671	1.472421945	838.2352941	102	101	1	0
DreamTeam109Agent	0.60008132	0.473751787	1.404410057	343.4705882	102	98	4	0
Agent007	0.52312451	0.435163615	1.307120148	263.6372549	102	92	10	0
CompromisingAgent	0.61568731	0.429626226	1.280832019	46.10784314	102	89	13	0
AgentFish	0.45845225	0.399028563	1.219032319	167.8627451	102	87	15	0
RGAgent	0.62195569	0.446108791	1.193267152	75.65686275	102	79	23	0
AgentFO2	0.49837477	0.414060155	1.129223106	286.3529412	102	76	26	0
Tjaronchery10Agent	0.47915006	0.326872134	0.99032478	53.7254902	102	70	32	0
LearningAgent	0.57617381	0.347110905	0.971548331	52.95098039	102	65	37	0
ChargingBoul	0.51195979	0.367583894	0.966812673	593.1764706	102	64	38	0
Agent4410	0.47320298	0.34283658	0.910276279	610.2352941	102	61	41	0
ProcrastinAgent	0.55670544	0.300984654	0.898214958	543.254902	102	62	40	0
LuckyAgent2022	0.49544612	0.319053643	0.877331995	398.5098039	102	59	43	0
MiCROAgent	0.50388242	0.316135958	0.856758715	854.3333333	102	57	45	0
BIU_agent	0.48238604	0.307864875	0.823583828	114.9607843	102	54	48	0
SuperAgent	0.42227981	0.272333047	0.728424824	692.9411765	102	48	54	0
GEAAgent	0.31327959	0.19310289	0.566671316	5.803921569	102	40	62	0
SmartAgent	0.28689619	0.127495878	0.419301663	69.88235294	102	30	72	0

Figure 3: Execution Result

4 Conclusions and Future Work

Although I did not use learning in this strategy, I was still able to create an agent that could agree on most negotiations by using time difference. However, learning and the opponent model are also important ideas in automatic negotiation, and I would like to use them next time.

Acknowledgement

I would like to thank Takayama Shota for useful discussion. Through discussion, I gained a better understanding and I could create a better agent.