Intelligent Agents – Agent Negotiation Coursework Report

Mortimer Sotom

mortimer.sotom@gmail.com

Worked in collaboration with Guilherme Barreiro Vieira for the agent code.

ABSTRACT

This paper presents an agent developed for a multi-lateral, multi-issue, time-based negotiation with uncertainty about the opponents' preferences with the aim to maximise its own utility while minimising the distance to the Nash equilibrium. Relevant information regarding the strategic aspects of negotiations is presented followed by an in-depth explanation of the adopted strategy. A frequency model takes into account the Nash product of bids to maximise the chances of an agreement while combining a time and utility based acceptance strategy to maximise utility and minimise Nash distance simultaneously. The competition results are critically analysed and highlight the high performance of the agent. Finally improvements are suggested to optimise the agent.

Keywords

Agent Negotiation; Acceptance Strategy; Nash Equilibrium; Frequency Analysis; Nash Product.

1. INTRODUCTION

This paper presents the negotiation strategy for the Southampton Intelligent Agents module competition. The competition is a multi-lateral, time-based negotiation which uses the stacked alternating protocol. The negotiation environment consists of multi-issue scenarios which include uncertainty about the opponents' preferences. The competition ranking is then based on an agent's own utility as well as the mean distance to the Nash equilibrium.

The agent will be described in terms of the chosen opponent model, the acceptance strategy and the bidding strategy. All decisions made and motivations for adapting strategies are explained at the beginning of each section. All variables which contribute to the acceptance and bidding strategy were tested for optimality with the results available in the appendix. Finally the results of the tournament are analysed to evaluate the performance of the agent with possible improvements.

2. STRATEGIC THEORY

Research performed by [1] found that the main opponent model approaches are frequency models and Machine Learning based models such as Bayesian, Neural Networks and surrogate models. In order to predict which model would be most effective, it is imperative to take into account the aspects of the negotiation tournament and the features of both modelling techniques.

Machine learning based models have proved to be more accurate than frequency models but have the constraint of performing time-consuming computations. [2] This computational time naturally increases with the degree of accuracy of the model and the domain size. Frequency models on the other hand are much simpler to implement at the cost of modelling accuracy but have a much lower computation time requirement. Source [1] expands on this time-exploration trade off by stating that "the gain in using a model should be higher than the loss in utility due to decrease in exploration of the outcome space." Since the negotiations are time-based, time remains a key factor. Furthermore the negotiations are multi-lateral so the computational time and memory used by a model is multiplied by the number of opponents. This might be negligible in small domain sizes but will increase exponentially in importance in larger domains.

3. AGENT DESCRIPTION

Before reaching a final decision, a review of the best performing agents of the ANAC competitions was studied. HardHeaded [3] used a frequency model with a multi-concession strategy and had the best overall performance. Johnny Black [4] obtained second place in the Nash distance category with a more advanced acceptance and bidding strategy. The change in concession strategy employed by HardHeaded and the acceptance strategy of Johnny Black were the main influences for this agent.

3.1 Opponent Modelling

To ensure there is no confusion in the explanation of equations, the basics of a bid are explained. A bid consists of a set of issues i with a weight w_i. Each issue has multiple options o with an evaluation value e_o. The set of options for an issue is noted as O_i.

3.1.1 Evaluation Function

The evaluation values of an option can be calculated using a logical and intuitive equation show below where, f_o is the frequency of option o and f_{oMAX} is the maximum frequency an option was used. This equation is used to calculate the normalised evaluation values of all options for each issue.

$$e_o = \frac{f_o}{f_{oMAX}} \qquad \text{(equation 1)}$$

3.1.2 Issue Weights

Modelling the issue weights is more complex so the opponent issue weight function by agent Johnny Black [4] was implemented. $\widehat{w_i}$ is the unnormalised weight of issue i and N_r is the total number of rounds or bids considered.

$$\widehat{w_i} = \sum_{o \in O_i} \frac{{f_o}^2}{{N_r}^2} \qquad \text{(equation 2)}$$

Once $\widehat{w_i}$ is calculated for all issues, the normalised weights of issues w_i are calculated as follows, where I is the set of issues:

$$w_i = \frac{\widehat{w_i}}{\sum_{i \in I} \widehat{w_i}} \qquad \text{(equation 3)}$$

3.1.3 Hard Headedness

As mentioned previously, many agents seem to adopt a Boulware tactic with a low conceding rate resulting in higher difficulties in reaching agreements. Therefore a variable is attributed a normalised value to express how hard headed, h, an opponent is. Δ_i represents the number of times an issue value changed and N_i is the total number of issues.

$$h = 1 - \frac{\sum_{i \in I} \Delta_i}{N_i * N_r} \qquad \text{(equation 4)}$$

This value is implemented in the acceptance strategy and will be explained in more detail in the following section.

3.2 Acceptance Strategy

There are four commonly used acceptance conditions [5] and this agent combines three of them for the acceptance strategy:

$AC_{const}(\alpha)$: a bid is accepted when the utility of a bid offered by an opponent is higher than a certain utility.

AC_{next}: a bid is accepted when the utility of a bid offered by an opponent is higher than the utility of the agent's proposed bid. This is implemented as it is in the acceptance strategy.

$AC_{time}(T)$: the opponent bid is accepted when a fixed amount of time has passed.

Source [6] explains the shape of the concession curve for traditional Boulware and Conceder tactics and states that a simplified Boulware utility acceptance strategy is:

$$AC_{const}(\alpha) = \frac{\log\left(\frac{(\min(t, Tmax))}{Tmax}\right)}{c(t)} + Ka$$

The logged nominator is essentially a function which expresses the amount of time left (normalised), which simplifies the equation to equation 5 below. *c(t)* and *Ka* are constants which represent a conceding factor governing the concession rate and the minimum starting utility respectively. Values for both of these were tested and explained in the next section.

$$AC_{const}(\alpha) = \frac{\log(t_{left})}{c(t)} + Ka \qquad \text{(equation 5)}$$

Finally, a variant of $AC_{time}(T)$ was applied in conjuncture with the hard headedness value *h* to determine the value of *c(t)*. Once 90% of the negotiation time has passed, the value of *c(t)* is decreased to switch to a conceding strategy. If the opponent is considered to be hard headed, the value of *c(t)* is further decreased to further accelerate the conceding rate and ensure an agreement is met.

$$c(t) = \begin{cases} 13 & if\ 0.1 < t_{left} \leq 1 \\ 10 & if\ 0 < t_{left} \leq 0.1 \land h \leq 0.6 \\ 7 & if 0 < t_{left} \leq 0.1 \land h > 0.6 \end{cases}$$

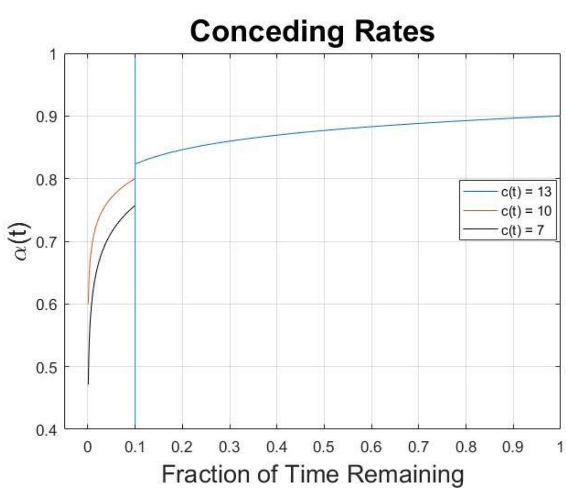

Figure 1: Concession curve of acceptance strategy

3.3 Bidding Strategy

It is common practice for agents such as TheNegotiator or BRAMAgent [7] to offer their maximum utility at the start of negotiations while it builds an opponent model and searches the outcome space. This assumption is used for the frequency model and followed by this agent to allow opponent to model us to increase chances of finding an agreement near the Nash equilibrium point. For the first 20% of the time, the agent offers a bid which maximises its own utility.

Every 10 rounds the Nash product of the best saved offer of opponents is updated to increase the accuracy of the opponent model by taking into account new offers.

Each round, 100 random bids are generated above the minimum utility from $AC_{const}(\alpha)$, and the bid with the best Nash product is saved in a bestBids list which starts with a size of 100. After 100 rounds, the number of bids generated and the size of the bestBids list is reduced by 5 every 100 rounds until they reach a value of 10 to reduce the time searching the outcome space and sorting the list in order of utility. If the bestBids list is full, the worst bid in the list is compared to the best randomly generated bid and only the best is stored to continually update the list.

Since the opponent model can only be expected to be approximate and it is used to calculate the Nash product of bids, one of the top 5 bids from the bestBids list is proposed.

4. TESTING

All variables mentioned in the previous sections were tested with different values to find the optimal one in order to increase the performance of the agent and maximise the chances of winning the competition. Before randomly testing these variables, a benchmark of which agents to compete against and in which domain had to be established. ConcederNegotiationParty and BoulwareNegotiationParty were quickly decided to be standard and easy agents to compete against. YXAgent came 2nd in the 2016 ANAC competition when agents would base their opponent model on repositories of past tournaments while Clockwork agent is more self-interested. Both of these agents were considered to be more challenging agents to agree with, especially in conflicting domains. The Party domain was chosen as a medium sized domain with 4 options per 6 issues.

For each variable tested, a balance between maximising utility and minimising Nash distance had to be found. The final chosen values can be seen as highlighted in yellow in appendix A. In multiple tests, the utility and Nash distance did not change or the changes were minimal so the median value was chosen.

5. TOURNAMENT ANALYSIS

The performance of this agent in the competition will be analysed in terms of utility obtained and distance to Nash equilibrium for each domain size. The domain sizes in table 1 are estimations by multiplying the number of options for between issues.

Overall the agent was successful in in maximising the number of agreements with 90.2% average agreement rate. A high average which is likely to be mainly due to the opponent model and bidding strategy which took into account the preferences of the opponents.

Table 1: Agent agreement across domains

	Domain 0 size 243	Domain 1 size 7200	Domain 2 size 23040	Total
Total negotiations	2758	2632	2840	8230
Number of agreements	2653	2455	2314	7422
% agreement	96.2	93.3	81.5	90.2

Boxplots enable the study of the distribution characteristics of a dataset of scores. Figure 2 presents the boxplots for all categories with the agent's score shown as a blue dot to easy visual interpretation.

In a small domain, most agents can be expected to explore the outcome space and identify bids near the Nash equilibrium point while agents with machine learning based models should be the most accurate. The high performance of this agent is then likely to be due to the Boulware tactic in the acceptance strategy and offering bids with a high Nash product.

The medium sized domain is where the agent performed poorly although the utility obtained is near the upper quartile meaning it remains in the top 25% of agents for that category. Machine learning based models were probably able to accurately model their opponents giving them the advantage of maximising their utility.

In regards to the large domain, again the agent's score is above the upper whiskers and is therefore considered to be a high outlier. In such large domains, a fast computational approach is more beneficial to explore the wide outcome space. However it is also the domain where the agent had the least percentage of agreements which can be due to a lower number of bids being exchanged between agents if one of the opponents is attempting a complex opponent model.

Generally, the agent did extremely well and obtained one of the top scores in the competition. It is important to note that although one aspect of the agent's strategy might be more suited for a certain category, it is the overall approach and how different strategical features complement each other that is responsible for the agent's success.

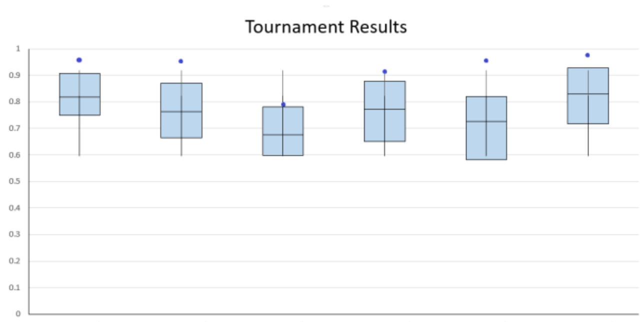

Figure 2: Boxplot of tournament results for the utility and Nash distance for all three domains.

5.1 Agent Improvements

Although the agent performed really well across all categories, some improvements could still be implemented. One such improvement is adding a machine learning based opponent modelling approach. At the beginning of the negotiation, once the agent reads its preference profile, it can approximate the domain size of the negotiation. If the domain size is judged to be smaller than some pre-defined value, the more accurate machine learning based model can be used without any drawbacks. For larger domains, the frequency model remains the better option.

Further testing could be carried out with a wider range of values to optimise the agent's performance. Similarly, multiple variables could be changed simultaneously to understand how they might affect one another and to identify winning combinations. However these are refinements more than improvements.

6. Conclusion

There are many factors to take into account while building an agent. Not only must the strategic aspects of negotiations be carefully selected, they must be tailored to the application or environment in which the agent will run. This agent was able to maximise its utility while minimising the distance to Nash equilibrium across multiple domain sizes as shown by the tournament results. Possible improvements were suggested but those tend to be refinements. The secret to an agent's performance is ensuring all strategic features implemented complement one another.

7. REFERENCES

[1] Baarslag T., Hendrikx M., Hindriks K., Jonker C. (2012) Measuring the Performance of Online Opponent Models in Automated Bilateral Negotiation. In: Thielscher M., Zhang D. (eds) AI 2012: Advances in Artificial Intelligence. AI 2012. Lecture Notes in Computer Science, vol 7691. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-35101-3_1

[2] Kattan A., Ong Y.S., Galvan-Lopez E. (2013) Multi-Agent Multi-Issue Negotiations with Incomplete Information: A Genetic Algorithm Based on Discrete Surrogate Approach. IEEE Congress Evolutionary Computation (CEC). DOI: 10.1109/CEC.2013.6557877

[3] Baarlag T. Fujita K., Gerding E. H., Hindriks K., Ito T., Jennings N. R., Jonker C., Kraus S., Lin R., Robu V., Williams C. R. (2012) Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition. Artificial Intelligence vol. 198 pp73-103. DOI: https://doi.org/10.1016/j.artint.2012.09.004

[4] Yuccel O., Hoffman J., Sen S.(2016) Johnny Black: A Mediating Approach to Mulitlateral Negotiations. Studies in Computational Intelligence, Modern Approaches to Agent-based Complex Automated Negotiation vol. 674 pp231-238 DOI: 10.1007/978-3-319-51563-2_18

[5] Baarslag T., Hindriks K., Jonker C. (2013) Acceptance Conditions in Automated Negotiation. In: Ito T., Zhang M., Robu V., Matsuo T. (eds) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol 435. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-30737-9_6

[6] Cao M., Dai X. (2014) Multi-Strategy Selection Model for Automated Negotiation. 47th Hawaii International

Conference on System Science. DOI: 10.1109/HICSS.2014.40

[7] Dirkzwager A.S.Y., Hendrikx M.J.C., De Ruiter J.R. (2013) TheNegotiator: A Dynamic Strategy for Bilateral Negotiations with Time-Based Discounts. In: Ito T., Zhang M., Robu V., Matsuo T. (eds) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol 435. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-30737-9_16

8. APPENDIX

Appendix A: Testing results

Note: This table should be common across all members of group 30 since the testing data was used to optimise the agent before submitting it for the competition.

	Agents	Utility	Nash distance
Reduce bestBids list			
Every 10 rounds	Easy	0.8637	0.3126
Every 50 rounds	Easy	0.8637	0.3126
Every 100 rounds	Easy	0.8692	0.3092
Reduce n#BidsGen			
Per 10 rounds	Easy	0.8637	0.3126
Per 50 rounds	Easy	0.8637	0.3126
Per 100 rounds	Easy	0.8637	0.3126
Reducing both			
Per 10 rounds	Easy – hard	0.86642 - 0.7665	0.2953 - 0.2358
Per 50 rounds	Easy – hard	0.8637 - 0.7550	0.31263 - 0.2510
Per 100 rounds	Easy – hard	0.86642 - 0.7882	0.30920 - 0.2009
Ka			
1	Easy – hard	0.9489 - 0.8371	0.3854 - 0.2318
0.95	Easy – hard	0.9154 - 0.7376	0.3002 - 0.2571
0.9	Easy – hard	0.8535 - 0.8007	0.2928 - 0.22
0.8	Easy – hard	0.7951 - 0.7338	0.2523 - 0.3064
hardHeaded h			
0.6	Easy – hard	0.8637 - 0.8089	0.3126 - 0.2082
0.5	Easy – hard	0.8535 - 0.8007	0.3162 - 0.22
0.4	Easy – hard	0.8631 - 0.7649	0.3126 - 0.2736
% time left before changing conceding factor			
95	Easy – hard	0.8637 - 0.8072	0.3126 - 0.22
90	Easy – hard	0.8535 - 0.8007	0.3162 - 0.22
80	Easy – hard	0.8620 - 0.7494	0.3126 - 0.2226
% time offering max utility			
10	Easy – hard	0.8637 - 0.7923	0.3126 - 0.1836
20	Easy – hard	0.8535 - 0.8007	0.3162 - 0.22
30	Easy – hard	0.8637 - 0.7820	0.3126 - 0.2562

Conceding factors			
7 to 6	Easy	0.8637	0.3126
7 to 7	Easy	0.8637	0.3126
7 to 8	Easy	0.8637	0.3126
10 to 9	Easy	0.8637	0.3126
10 to 10	Easy	0.8637	0.3126
10 to 11	Easy	0.8637	0.3126
13 to 11	Easy	0.8568	0.3111
13 to 12	Easy	0.8603	0.3119
13 to 13	Easy	0.8637	0.3126
13 to 14	Easy	0.8712	0.3157
13 to 15	Easy	0.8739	0.3148