Intelligent Agents: Agent Elman

Mohammad Ali Khan
University of Southampton
mak1g11@soton.ac.uk

ABSTRACT

This paper discusses the TAC Trading Competition, mentioning different challenges faced. It then details the design and strategies of Agent Elman, justifying the decisions taken. Lastly, it analyses the result of the competition and the agent's achievement of seventh position.

Categories and Subject Descriptors

D.3.3 [**Intelligent Agents**]: TAC Trading Agent Competition – *intelligent agents, trading competitions.*

General Terms

Algorithms, Measurement, Documentation, Performance, Design, Economics, Reliability, Experimentation,

Keywords

agent, trading, competition

1. INTRODUCTION

TAC (Trading Agent Competition) is a well-known competition used as a competitive benchmark for intelligent agents. It consists of a number of clients needing travel packages for a vacation over a notional 5-day period. They need to travel to Tampa, stay at a hotel and possibly attend entertainment events during their stay. Each game lasts 9 minutes, consisting of 8 agents; they have to buy flights, hotel rooms, and trade tickets to complete the best packages for their clients, the score being package cost subtracted from package utilities. These games are run numerous times and the average score is taken for an accurate representation of agent performance.

2. DESIGN

We firstly set made the code a lot more readable. To achieve this, we created *Client*, *ClientPackage* classes to keep track of different packages we created for our clients and also checking if they are feasible or not. We also created a *Tracker* class to keep track of different tickets that our agent had for different clients.

During the course of the competition, our agent would initalise our classes and update the different values with time. Near the end, we check the feasibility of our packages, making amends to complete packages if they were infeasible.

2.1 Flights

There are no restrictions to number of flights available each day, the cost of the tickets initially being $250 – 400 and then stochastically perturbing between $150 – 400. Flight prices generally increase towards the end of the auction [1], meaning it's good to buy tickets as soon as possible, allowing us to get them for a cheap price. However, we ran a number of simulations of the function to observe the tickets and discovered that the best time to buy flights would be just about halfway through the competition. The figure below shows the different bounds with different starting values of x

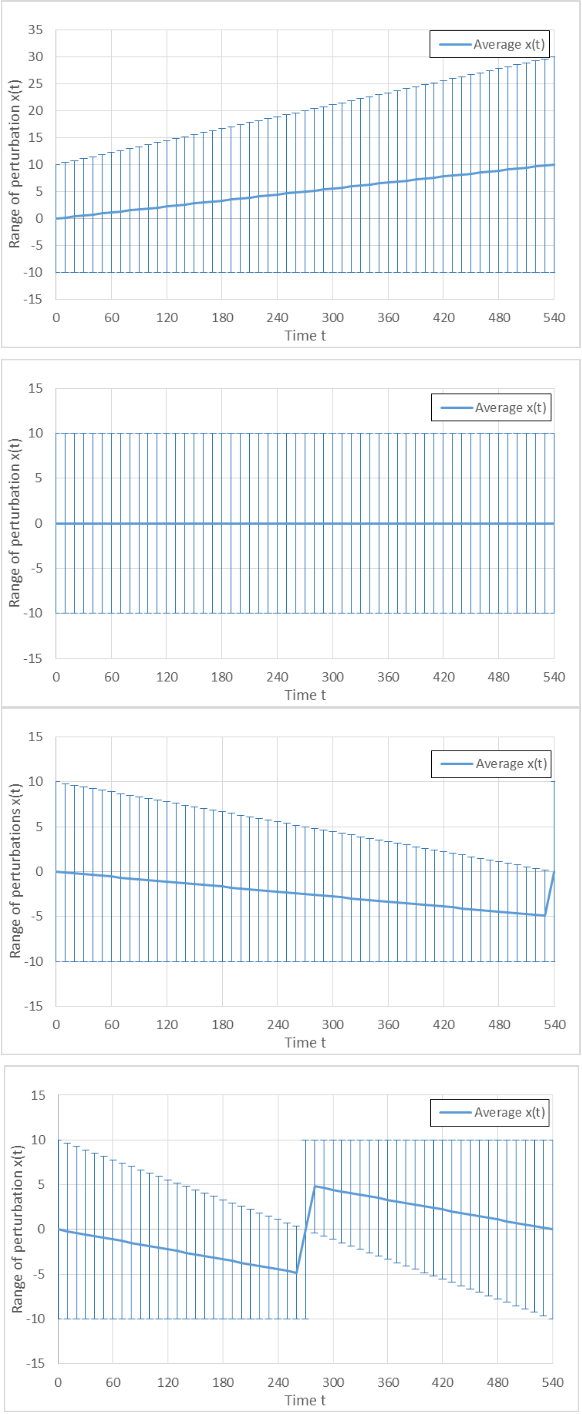

Figure 1. Bounds for function as time progression with different values of x (i) x = 30, (ii) x = 10, (iii) x = 0, (iv) x = -10)

Therefore, we decided to be more opportunistic in our bidding strategy by following the perturbations of the function. If ticket prices increased by more than $10 in the next perturbation, our agent quickly buys the ticket, as the function is an increasing one. Otherwise, it would wait for the price to lower, although automatically buying at $150.

2.2 Hotels

Hotel auctions, an implementation of a Vickrey auction [2], are the most important auction. The limited amount of hotel rooms coupled with different combinations of client preferences means our agent could potentially miss out on the hotels if our hotel bids aren't good enough. This would also render our package void as clients need to stay at a hotel during their stay and cannot switch hotels. Therefore, we developed many strategies to improve our agent, the simplest being bidding $251 initially as opposed to $250, giving us an edge over naïve agents.

2.2.1 Deciding hotel type

Choosing hotel type for our client depended on two factors: utility and duration of stay. The good hotel would cost more most of the time and we do not want to overspend especially if we are not getting good utility out it. Moreover, getting good hotel rooms for more days would be difficult and we are likely to miss out a room and jeopardise our package. Our agent bids for good hotels for clients having utility of more than 90 for Tampa Towers and duration of three days or less

2.2.2 Reacting on competition

The naïve agent strategy was to always add 50 to the agent asking price. This meant that the increase in agent bidding was static. Instead, we decided to calculate the different between the last two ask prices. This way, we are reacting on the competition and have a better chance of getting what we need.

2.2.3 Rescheduling

To mitigate the problem of missing out on a hotel room and voiding our package, we improved the naïve plan of the dummy agent. Once all hotel auctions close, we loop over our packages and check any of them is infeasible. In such a case, our agent figures out the next longest package possible with the available rooms and purchases an extra flight to or from the vacation venue. We take a small hit with the extra cost of the flight but complete our package.

2.2.4 Limit

Another problem is overbidding. This is when some agents bid high on some tickets; our agent would try to compete, but we did not want to spend too much money. Therefore, we also enforce a limiting price of $650, so that we do not end up going in loss.

2.2.5 Scatter Shot

We also bid for a hotel in an auction that we are not participating in (i.e. we do not need any rooms from that auction). In this case, our agent makes a small bid of $20. If we get the room, we get it for a cheap price and it could possibly be used in a case where a client is coming for a stay for a single day. If we do not receive the hotel room, this means we still managed to raise the selling price of room, a win-win for us.

2.2.6 Possible Improvements

It would have been nice to write a proper optimiser for the hotel auctions. A potential improvement on our current strategy is to change the hotel type for a client if we fail to get a hotel room for the initial type. This would do well if the client just has a single night stay. However, we would need to overcome the challenge of having more than one hotel room to book, as well as checking if the auctions for the hotel rooms we need (of the new type) haven't closed either.;

2.3 Entertainment

Agents can buy and sell entertainment tickets. This means that the agent are looking to obtain high utility tickets from other agents low prices while selling unwanted tickets for the highest price.

2.3.1 Selling strategy

Generally, the best value for money price to sell tickets is between 80 and 60 [3]. With this in mind, we start a function to start selling at 130 initially, slowly decreasing our price over time till it costs 85, the lowest we go. We do not go lower as that would give an advantage to other agents as it's better to keep these tickets than increase a competitor's score.

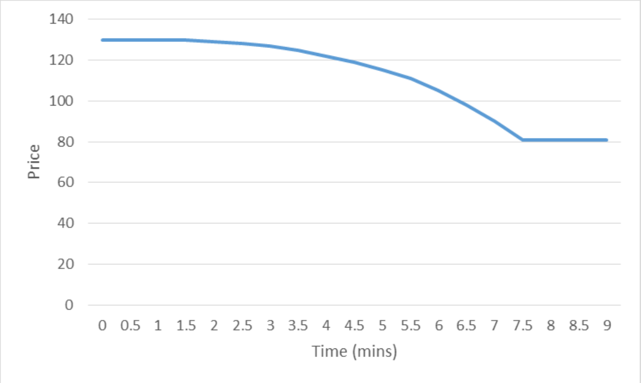

Figure 2. Decrease of entertainment price with game time

2.3.2 Buying strategy

We do not want to pay more than our utility for a ticket. Therefore, we start from a low bidding price for tickets that we need, slowly increasing our bidding price until it is slightly less than the utility of the ticket type. This prevents overspending on entertainment tickets.

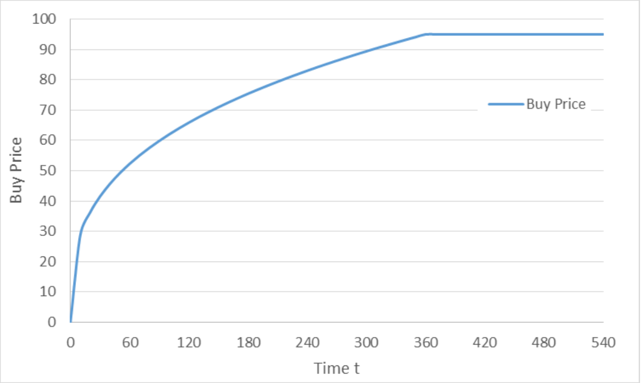

Figure 3. Increase of entertainment price with game time (for item with utility of 100)

3. ANALYSIS

3.1 Position

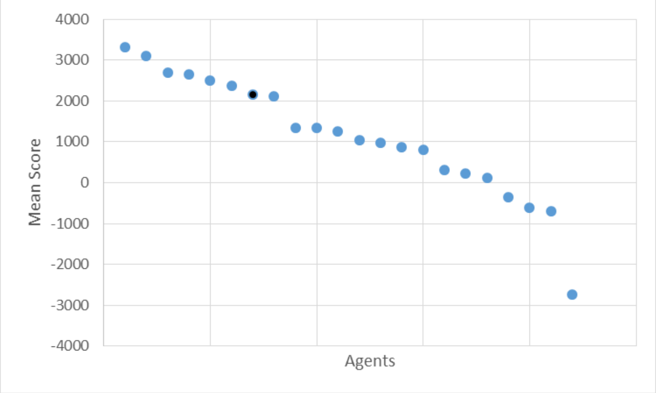

Figure 4. Scatter plot of different agent mean scores (AgentElman is black)

AgentElman finished a 7th in the competition with an average score of roughly 2154. This was in a small pocket range of scores from 2100 – 2700, consisting of six agents. The pocket above this consisted of two agents with scores over 3000 and the one below was a drop to scores less than 1400. During the course of the 16 games, AgentElman had a negative score in only one game, while also scoring less than 1000 only 5 times. Regarding positions in each game, AgentElman never went below 4th position, achieving an average of 2.56 over competition. If the competition was ranked by average game position, AgentElman would come 3rd.

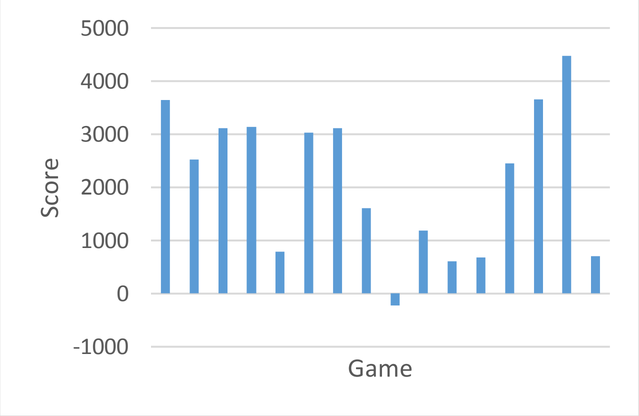

Figure 5. Bar chart showing AgentElman scores

3.2 Proactivity vs. Reactivity

It was interesting to view the averages of different agents over the course of competition. We recorded the readings of agents from 1st to 10th position to see how the agents progressed.

As we follow the progress the agents, we see that our agent started brightly, before a slow decline in performance. Other agents starting at a score similar to AgentElman mostly remained around the same area while almost all the others in the 2000 pocket above AgentElman improved during the competition, surpassing our agent to a higher average score. It is probable that these agents were worked on during the day of competition and their strategies improved. We decided against working to optimise our agent during the competition as we were afraid we might unintentionally break it, causing a decline in performance. Moreover, our agent, while not pushing the top scoring ones, seemed to be performing well. This shows that while our initial strategy was pretty good, we did not react to the proceedings of the competitions, which could potentially have given us a higher average score and position.

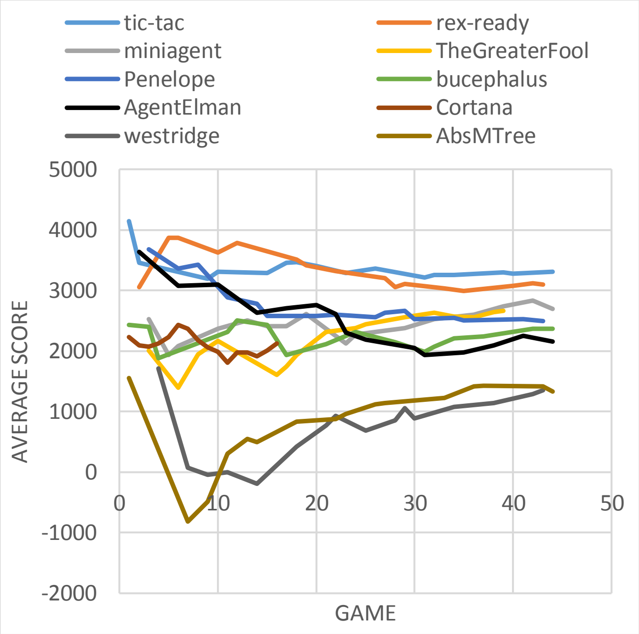

Figure 6. Average scores of first ten agents as game progressed

4. CONCLUSIONS

AgentElman had a solid design and employed a number of interesting strategies for the different auctions, resulting from self-experimentation and reading literature. As it can be seen from the results, Agent Elman did well above average, achieving 7th position. However, it must be noted that the bidding strategy for hotel auctions could have been further improved and optimised. Moreover, had we been more reactive on the day of the competition, AgentElman could have been higher on the table.

5. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

6. REFERENCES

[1] A. Kothari and B. Ferguson, "Blazin - A TAC Classic Agent".

[2] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders," *The Journal of finance*, vol. 16, no. 1, pp. 8-37, 1961.

[3] S.-F. Cheng, E. Leung, K. M. Lochner, K. O'Malley, D. M. Reeves, L. J. Shvartzman and M. P. Wellman, "Walverine: A Walrasian Trading Agent," *Second International Joint Conference on Autonomous* , p. 23, 2003.