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, Management, 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 trading agents. It consists of a number of clients who need travel packages for a vacation over a notional 5-day period. They need to travel to the venue, stay in a hotel and possibly attend entertainment events during their stay. Each game consists of 8 agents and lasts 9 minutes. In these games, agents have to buy flights, hotel rooms, and trade tickets to complete the best packages for their clients, receiving most utility for the least cost. These games are run many 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.

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 peturbing between $150 – 400. However, flight prices generally increase with time, meaning it's good to buy the ticket as soon as possible. It would allow us to get the flight tickets for a cheap price and get the flight purchase out of the way. However, it is still possible that sometimes, flight tickets would be more expensive at the beginning and hence cost us more in the long run. Moreover, they could also decrease in price to $150, which would be at least $100 even if we bought the flights at $250 initially, the lowest possible price. Therefore, we decided to be more opportunistic in our bidding strategy by following the peturbations of the function. If an ticket prices increased by more than $10 in the next perturbation, our agent quickly buys the ticket, as the function is an increasing on. Otherwise, it would wait, although automatically buying at $150 if the price fell to that.

2.2 Hotels
Hotels are the bottleneck of the whole competition and the single limiting factor. One of the reasons for this is the limited number of hotel rooms. This means that, given the different client preferences and combinations, our agent could potentially miss out on the hotels if our hotel bids aren't good enough. This also means that our package becomes void as the client needs to stay at a hotel during their stay and cannot switch hotels. Therefore, we developed many strategies we placed 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 on hotel
When it came to deciding which type of hotel our agent would bid on, we considered 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 day would be hard and we are likely to miss out a room and potentially 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 if none are infeasible any more. 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 to complete feasible packages.

2.2.4 Limit
Another problem is overbidding. This can be in the case where some agents end up bidding high on some tickets. Our agent would try and increase on that, but we do not want to end up spending too much money, sending us into a negative score. Therefore, we also enforce a limiting price of $650, so that we do not end up going in loss.

2.2.5 Scatter Shot
One last thing we do is 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.

2.3 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. Generally, the best value for money price to sell tickets is between 80 and 60. 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 and it's better to keep these tickets than increase a competitor's score.

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 equals the utility of the ticket type. This prevents overspending on entertainment tickets.

3. ANALYSIS

4. CONCLUSIONS

As it can be seen from the results, Agent Elman did well above average.

4.1 Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

4.1.1 Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

4.1.1.1 Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

4.1.1.2 Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

5. ACKNOWLEDGMENTS

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

6. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[2] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= http://doi.acm.org/10.1145/332040.332491.

[4] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.

[5] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/964696.964697.

[8] Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= http://dx.doi.org/10.1016/j.jss.2005.05.030.

[9] Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= http://doi.acm.org/10.1145/90417.90738.

Columns on Last Page Should Be Made As Close As Possible to Equal Length