

TAC Game Report

Tristan Claverie
950418P612
trcl16@student.bth.se

Damien Duvacher
XXXXXXXXXXXX
dadu16@student.bth.se

INTRODUCTION

In computer science, an agent is a software who works autonomously. It acts like an automaton according to how it was developed.

It's a kind of artificial intelligence.

An agent can be :

- Reactive: It can act regarding to his environment.
- Proactive: It can take initiative to reach its goal.
- "Sociable": It can communicate with other agents.

Agent can be used in many fields, for example there is an agent to organize import and export of containers in a container terminal at Göteborg, this agent permits to fill, empty and stack containers from ships in the dock via machines. Machine moving are managed by agents and are optimized to be the quickest as possible.

Another example very much simple is just a reactive agent which can send automatic responses when you order a command or something else.

So, agents are useful to do some tasks and can make life easier.

The TAC agent competition is an international forum designed to promote and encourage high quality research into the trading agent problem.

In the TAC classic, we are an agent responsible for 8 clients who want to go on holiday. We have to create travel packages to answer to their preferences (Flight date, Hotel preferences, Entertainments asked).

We have to participate to different auctions to make the packages. There are 3 different auctions:

- Flight auctions
- Hotel auctions
- Entertainment auctions

We will describe each of behaviour auctions in the rest of this report.

We will participate to a competition between each agent of the different group of students.

At the end of the competition we will have a score regarding our holding and the higher score will win.

So we have to design an agent to win this competition, this report will describe how we've developed our agent.

In this part we'll talk about what we've done before starting the development of our agent and what was our first thinking about the project.

Our first step was to analyse the dummy agent code to see how it works and how we can adapt this code for our agent. First we will describe the different auctions:

- Flight auction: for this auction, a price is set at the beginning of the game between (250 and 400) and then this price changes during the game (rise or decrease). There are no limited places for one flight so the only thing we have to do is decide when we buy our flight.
- Hotel auction: this auction is an English auction. We put a bid with a price and when the auction is closed, the ask price becomes the 16th higher price and then all the bids with a price over the ask price (according to the stock: there are 16 places available for each hotel and each night) are validated and players get the rooms. So to be sure to get an hotel we have to put an enough high price.
- Entertainment: players can send bid to sell or buy entertainments if some bids match together, the transaction is done.

For the flight auctions, the dummy agent buys everything at the beginning of the game, it buys all it needed. That can be the best solution if by luck all the auctions will grow but it's not always the case so we have to think about another strategy.

For the hotel auctions, the dummy agent updates his bid each time the quote is updated. He puts a new bid, the price of this bid is equal to the ask price + 50. It's not a really good strategy because 50 is a fix variable and variation of the ask price can be higher or lower in different games. Furthermore maybe this price is not enough high to get the hotel needed.

And for the entertainment, the dummy agent tries to sell entertainments that it doesn't need at the higher price and it tries to buy what it needs at the lower price.

For our agent, we decided to adopt a global strategy which is to get the best packages regardless of the price. With this strategy we will have a good score in satisfaction and the score about the price depends on the game and on the other players. Now let's present our strategy in details.

I. STRATEGIES

A. Flight strategy

For the hotel auction strategy, we've decided to put 2 limits:

- A low limit: if the price of the hotel that we want goes under this limit, we buy it.

- A high limit: if the price become too high and goes over this limit we buy it too, to prevent a too big spending.

At the beginning we just put a low limit, but after many tests, we saw that the price usually become really big after some times then, we decided to put a high limit. In a first step we've put some fixed limits (200 for the low limit and 400 for the high limit) but after thought, we realized that was not really relevant because the prices begin between 250 and 400 so there can be a gap between some auction.

So we decided to adapt the limits for each auction. We've put different limits for the each auctions, regarding the starting price of the auction:

- Low limit = Starting price - 10
- High limit = Starting price + 10

This is a better strategy than before because we adapt our price according to each auction and not only a global strat for all the auctions.

B. Hotel strategy

For the hotel strategy, it's a bit more complicated than the flight one.

As these auctions are english auction, we thought that we have to put a enough big price to get ou hotel but not to big to not make the price grow too much.

So this strategy is based on a variable "delta". This delta is the average ask price changes, each time the auction change, we update the delta by this function: $\text{delta} = (\text{delta} + \text{newAskPrice} - \text{oldAskPrice})/2$. With the function we can have an overview of what will happen in the future rounds. Each time the quote for the auction is update, we compared the ask price with our bid and then there is 2 possibility:

- The ask price is higher than our bid, in that case we change our bid price for the ask price+delta.
- The ask price is lower than our bid in that case we change nothing

This strategy permits to have a bigger price than the ask price because each time the ask price go over our bid we update it. And it permits to not have a too big price because if our price is higher than the ask price we don't update our bid.

Updating with the delta permits to predict the future growth of the price and allow us to not update our price each time there is a new ask price but just in case we will not have the hotel.

C. Entertainment strategy

II. STRATEGIES

A. Flight strategy

For the hotel auction strategy, we've decided to put 2 limits:

- A low limit: if the price of the hotel that we want goes under this limit, we buy it.
- A high limit: if the price become too high and goes over this limit we buy it too, to prevent a too big spending.

At the beginning we just put a low limit, but after many tests, we saw that the price usually become really big after some times then, we decided to put a high limit. In a first step we've put some fixed limits (200 for the low limit and 400 for the high limit) but after thought, we realized that was not really relevant because the prices begin between 250 and 400 so there can be a gap between some auction.

So we decided to adapt the limits for each auction. We've put different limits for the each auctions, regarding the starting price of the auction:

- Low limit = Starting price - 10
- High limit = Starting price + 10

This is a better strategy than before because we adapt our price according to each auction and not only a global strat for all the auctions.

B. Hotel strategy

For the hotel strategy, it's a bit more complicated than the flight one.

As these auctions are english auction, we thought that we have to put a enough big price to get ou hotel but not to big to not make the price grow too much.

So this strategy is based on a variable "delta". This delta is the average ask price changes, each time the auction change, we update the delta by this function: $\text{delta} = (\text{delta} + \text{newAskPrice} - \text{oldAskPrice})/2$. With the function we can have an overview of what will happen in the future rounds. Each time the quote for the auction is update, we compared the ask price with our bid and then there is 2 possibility:

- The ask price is higher than our bid, in that case we change our bid price for the ask price+delta.
- The ask price is lower than our bid in that case we change nothing

This strategy permits to have a bigger price than the ask price because each time the ask price go over our bid we update it. And it permits to not have a too big price because if our price is higher than the ask price we don't update our bid.

Updating with the delta permits to predict the future growth of the price and allow us to not update our price each time there is a new ask price but just in case we will not have the hotel.

C. Entertainment strategy