AUCTIONING AGENT

Liang Zixuan, Wang Junxiong 12/02/2015

Bidding Strategy

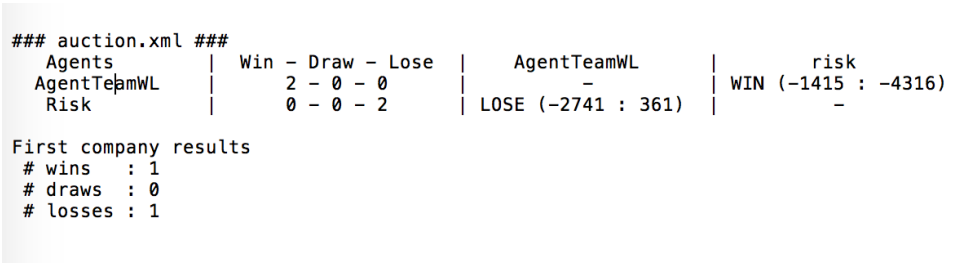
As we are supposed to operate a multi-vehicle company and competes against opponents in auction in this assignment, the implementation consists of two steps.

Planning

The first step is to implement the planning algorithm using which we can calculate our marginal cost and from whose output we can make a plan that we return to the auction house in the end. Since the algorithm from the previous assignment suited these needs, we reused it in this solution. Also since tasks are added one by one to the plan in each of the rounds, in order not to calculate the whole SLS algorithm from start each time, we insert the new task to old plan and find the best insertion location to construct our new plan, and calculate estimated marginal cost correspondingly.
After this part is done, one can participate in an auction by bidding the marginal cost (or additionally improve it by making some sort of a formula, including marginal cost). But taking into account only calculations about its own company and plan will probably not lead to a great performance against other competitors. That's why there is a second step, which is bidding strategy refinement.

Bidding

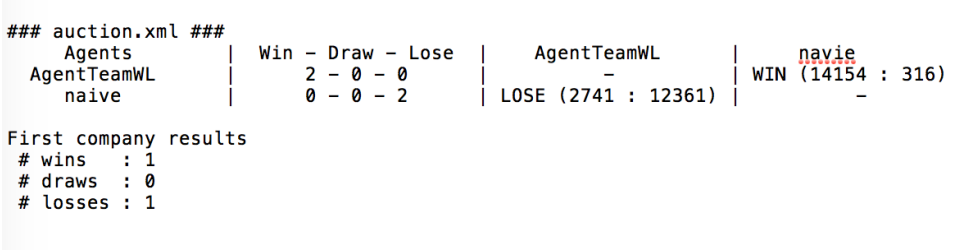
In this step we refine our strategy using information available to us through the system. This information is previous opponent bids given to us by $auctionResult()$, which we store and calculate several different things using them. We focus on deciding our bid according to estimated opponent marginal cost. The algorithm we use to calculate estimated opponent marginal cost is the same as that used to calculate our marginal cost, which is finding the best inserting location for the new task. Since this is a closed-bid auction, in order to bid better and be more competitive, we need to adapt to opponent's previous bidding behavior. If the situation allows, we just bid to win as many tasks as possible, with bid price as high as possible (at least not losing too much profit).
The strategy also relies on the assumption that it is better to bid less in the beginning, in order to acquire more tasks, be flexible in the future. This can be in a sense be looked at as "warming-up" company.
From bidding history we can know about tasks won by opponents, so we can calculate its marginal cost for a new task. After finding the marginal cost for the opponent, we multiply



```
### auction.xml ###
   Agents         |  Win - Draw - Lose  |    AgentTeamWL       |        risk
  AgentTeamWL     |      2 - 0 - 0      |          -           | WIN (-1415 : -4316)
   Risk           |      0 - 0 - 2      | LOSE (-2741 : 361)   |          -

First company results
 # wins   : 1
 # draws  : 0
 # losses : 1
```

Figure 1: Our Implementation vs Risky Implementation



```
### auction.xml ###
     Agents         |  Win - Draw - Lose  |    AgentTeamWL       |     navie
  AgentTeamWL       |      2 - 0 - 0      |          -           | WIN (14154 : 316)
     naive          |      0 - 0 - 2      | LOSE (2741 : 12361)  |          -

First company results
 # wins   : 1
 # draws  : 0
 # losses : 1
```

Figure 2: Our Implementation vs Risky Implementation

it by $ratio$. At that point we set our bid to be 75% of this calculated marginal cost. If that price is below certain limit, we correct the price to the lowest possible viable value. This way we are protecting ourselves from making too harmful bids. The bottom limit is calculated as $myMarginBidRatio * myMarginalCost$, which is set to be 0.8.

At this point our bid is not settled yet. It is worth mentioning that it is likely that a new task's path overlap with old plan, which means the marginal cost would be 0. Of course it is meaningless to offer free delivery for this new task, but we also cannot let it be easily won by the opponent. We believe that normally the opponent will set a minimum bid value for these zero-marginal-cost tasks. Thus if our bid is less than the opponent's minimum history bid which we denote as $bidOppMin$, we adjust our bid to $bidOppMin - 1$, so as to obtain higher profit while keeping good chance of winning the task.

As for the "company warming up" phase, we multiply our bid by a factor of 0.5 for the first four tasks, in order to grab more tasks as quickly as possible, and adjust our strategy in the future bids.

Obtained Result

The implementations against which we ran our implementations are: agent that are very risky and always bid a very low price and naive agent that just decide bid based on its own cost. As we can see from Figure 1 and Figure 2 our implementation won both matches.