Exercise 5: An Auctioning Agent for the Pickup and Delivery Problem

Group №34: Anh-Nghia Khau, Stephane Cayssials

November 27, 2016

1 Bidding strategy

Suppose that in this assignment, the optimal plan is generated by the algorithm that we used in the 4th assignment and the others agents use as well this algorithm (if they do not, they will always having a non-optimal plan in which giving a higher bid) We also suppose that for each round, the bid is obtained by multiply the marginal cost with a ratio and this ratio have to be in an defined interval.

For the bidding strategy, we bid using these information in **auctionResult** method. We try to estimate the bid of the opponent and then we give our bid in terms of the opponent bid. And we will try also take as many as the task possible in our strategy. So for each task in the auction, we add this task to the list of actual task and compute the marginal cost for our agent and the opponent. After we multiply each marginal cost with a ratio (ratio is a value that we update for each round depend on the previous bid) and return the minimum between these two values. Roughly, we have myBid = myMarginalCost*myRatio and opponentBib = opponentMarginalCost*opponentRatio.

For the first round, we can not guess anything and just return the marginalCost multiply by a ratio initial (0.85). At the second round, we have the bid of the opponent at the first round and we try to guess the **homeCity** of the vehicle that the opponent use (since we use the same algorithm and we also know every task that the opponent won). If we know the homeCity of the vehicle, we can compute the marginalCost of the opponent and we can as well compute the **ratio** that the opponent uses at the beginning.

How do we update the ratio of our agent and the opposant ?
So we have the ratio of the opponent at the beginning. And now, each time we receive the result we will try to update the ratio (the ratio have to be in an interval defined at the beginning). There is two cases:

- If we won, we update our bestPlan as the plan that we computed in the **askPrice** method. We can also suppose that the ratio that we have now is good and we try to increase (in addition with a small number) for having a bigger bid (bigger profit) for the next task at the auction. At the same time, the opponent know that he lose because his bid is too high and he will also subtract his ratio with a small number as well.
- If we lose, we will update the bestPlan of the opponent and this time we will do the compute opposite with the compute when we won (i.e. we will subtract our ratio and addition the ratio of the opposant with a small number).

```java
// if we win
 myRatio += myRatio*((double)myBid / (double)opponentBid);
 opponentRatio -= opponentRatio*((double)myBid / (double)opponentBid);
// if we lose
 myRatio -= myRatio*((double)opponentBid / (double)myBid);
 opponentRatio += opponentRatio*((double)opponentBid / (double)myBid);
```

2 Results

2.1 Experiment 1 with the auction random

2.1.1 Setting

In the first experiment, we just try to update the ratio by addition/subtract with a fraction previousMyBid/previousOpponentBid (without predicting the homeCity)

2.1.2 Observations

We won 6 tasks among 20 tasks, it's right because we do not have the right ratio of the opponent, so it's difficult to guess the bid of the opponent and we lose. $\vdots$

2.2 Experiment 2 with the auction random

2.2.1 Setting

In the second experiment, we try to guess the homeCity

2.2.2 Observations

We won 16 tasks among 20 task, it's not bad and it seems that we have guessed the right ratio initial of the opponent.