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

Group №17: Gervaise Lara

November 19, 2019

1 Bidding strategy

The planning algorithm based on the SLS algorithm from the previous assignment calculate the marginal cost and make a plan accordingly. Then, we return this plan to the auction house. As tasks are added sequentially at each round, we find the best location to insert the new task in the old plan to construct the new plan in order not to always calculate the whole SLS algorithm from scratch.

Then, it's time to participate to the auction by bidding the marginal cost! But one needs to take into account the competitors. It's where we need to refine the bidding strategy using the information available to us through the system and adapt to the opponent previous bidding behavior. Through *auctionResult()* we get the information about the previous bids and use it to compute the competitor marginal cost (using the same algorithm as for our agents).

We set our bid to $oppRatio \cdot oppMarginalCost$ and if it's below our lower limit $marginBidRatio \cdot myMarginalCost$, we adjust the price to the lowest possible viable value in order to not make stupid bids.

We also need to take into account that a new task path may overlap with the old plan, resulting in a marginal cost of 0. On one hand, it's meaningless to offer free delivery for this new task, but on the other hand, we cannot let the opponent winning it easily. Believing that the opponent would set a minimum bid value for these zero-marginal cost tasks, we adjust our bid to the opponent minimum bid in the history *bidOppMin* : if our bid is below this value, we set it to *bidOppMin*-1. This allows us to obtain a high profit while still being in competitive to win the task.

In addition to that, the strategy relies on a simple principle: at the beginning, we bid less (by multiplying our bid with a factor $initialBidRatio = 0.5$) in order to be flexible in the future. Then, we bid with the aim to win as many tasks as possible, with a bid price as high as possible without losing too much profit.

Finally, we update at each round the *oppRatio* and the *marginBidRatio* we use for our calculation according to the outcome of the previous auction : if we won we increase them, if we lost we decrease them.

2 Results

2.1 Experiment 1: Comparisons with dummy agents

2.1.1 Setting

We run the experiment with 2 vehicles per company with a uniform distribution of 5, 20 and 50 tasks on the England topology comparing our agent to :
- a risky agent which always bids at a very low price
- a naive agent which bids only based on its own cost

2.1.2 Observations

Even if the risky agent performs better, we won all the tournaments for all the different number of tasks.

However, given that our agent has a "warm up" strategy which consists in bidding at a very low price for the 4 firsts rounds, it doesn't perform well on a very low number of tasks (below 9 tasks).

We should also mention that the agent "time out" if there are more than 50 tasks. This is because following the information given about the tournaments, we ensured to adapt the number of iteration and thus the cost of the SLS algorithm only for a number of tasks below 50.

2.2 Experiment 2: variations of the internal parameters

2.2.1 Setting

We keep the England topology with 2 vehicles per company, vary the number of tasks and compete against the Naive agent. We vary some parameters described previously:
- *marginBidRatio*
- *oppRatio*
- *initialBidRatio*
- *NB FIRST BIDS*

2.2.2 Observations

Starting with $marginBidRatio = 0.8$ and $oppRatio = 0.85$ and fixing $initialBidRatio = 0.5$ and $NB\ FIRST\ BIDS = 4$ are the optimal parameters : in general, higher or lower values lead to a decrease of the overall performance. However, decreasing *initialBidRatio* or *NB FIRST BIDS* on a low number of tasks lead to better results.