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

Group №27: Carolin Beer, Francesco Pase

25th November 2018

1 Bidding strategy

Our bidding strategy consists of two main components. Firstly, we use a multitude of different estimation strategies (SLS, moving average, moving median, linear regression) to get an approximation of the bid that the opponent is going to place. Secondly, we utilize the probability distribution of tasks to compute the probability that there will be a task auctioned which has 0 marginal costs for the agent.

The feedback from previous auctions is used to adjust the weights of the different opponent estimation strategies. While every strategy is initially assigned the same weight, over time, strategies with a lower error will be assigned a higher weight and the other way around. As the competition goes on, the adaptions of the weights will become smaller and smaller. Furthermore, the different strategies themselves use the feedback value for adjusting their computations, e.g. the linear regression is refitted after every round.

All this information is aggregated to a bid. We compute our own marginal costs through SLS and ensure that our computed marginal cost are in between 0 and the theoretical maximum of the marginal costs (at most, the marginal costs for a task are $2 * max_{c,d \in C} distance(c, d)$). Moreover, when a task is auctioned, in order to compute the marginal we evaluate a new plan that includes such task. From this, using the task distribution, we compute the probability that the next task to be auctioned will have marginal zero (falls exactly in two cities we have to visit, in order). This information allows us to compute a "secret" utility this task has for us. We then combine these marginal cost that we computed along with the information about the estimated opponent bid and the probability that we will be able to pickup tasks that don't incur marginal costs. However, we make sure, that the estimated opponent bid won't pull the bid below our marginal costs.

Additionally, there is a 'warm up' period where we are willing to carry out a task for less than our marginal costs. The reasoning behind this is as follows: once we already have some tasks, the marginal costs for further costs are likely to decrease (as we can integrate them intelligently into our plan with SLS). This way, we avoid being 'stuck' without a single task due to comparatively higher marginal costs.

2 Results

2.1 Experiment 1: Testing different factors to incorporate opponent estimation

2.1.1 Setting

In the first experiment, we test how our agent performs when incrementing our bid by the (positive) delta between the opponent estimation and our bid, utilizing to different factors:

$$bid = bid + factor * (E(bid_{opponent}) - bid)$$

We compare a factor of 0.5 and a randomized factor following a U[0.3,1] distribution. We us the topology of England with 15 auctioned tasks. The timeouts for bids and plans are both 3030ms.

We implemented dummy agents representing bidding strategies of (i) the dummy agent given in the template, (ii) a dummy agent with constant bids (1000 and 1500 are used) and (iii) an SLS agent bidding its marginal costs and use them to assess our agent.

2.1.2 Observations

Factor	U[0.3,1]	0.5
Template dummy	(4012.5, -2398)	(1229.5, -3930)
Constant dummy (1000)	(-288, -428)	(-603, -1989)
Constant dummy (1500)	(12363, 0)	(16131, 0)
SLS dummy	(88235, -0.5)	(43480, -0.5)

Table 1: Comparison of the profit for different factors against the dummy agents. The tuple denotes the profit of our agent vs the profit of the dummy agent values

As we can see, the agent wins all the competitions, regardless of the factor used. However, since the randomized factor tends to lead to larger profits, we will continue to use it for the following experiment.

2.2 Experiment 2: Assessing the agent for different numbers of auctioned tasks

2.2.1 Setting

In this experiment, we test the chosen configuration for different amounts of auctioned tasks and test it against the dummy agent given in the template. The topology and timeouts remain the same.

2.2.2 Observations

#Tasks	Template dummy
10	(6380.5, 11)
14	(5186, 1048)
20	(16228, 3054)
40	(33828, 21675)
50	(66148, 23325)

Table 2: Comparison of the profit for different amounts of tasks against the dummy agents.

Especially with few tasks, the template dummy barely winds any task in the competition. With higher number of tasks this changes slightly as the bids of the template dummy are positively correlated with the id of the task (and thus increase along with the number of tasks). However, our agent still wins in all of the above settings.