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

Group №5: Anay Mehrotra, Anubhav Mittal

November 16, 2019

1 Bidding strategy

We start by bid aggressively to acquire initial tasks and become more cautious with every task acquired. We expect to gain a competitive advantage by aggressive bidding, and then compensate incurred costs. Given a task t, we look at the decrease in the expected cost of delivering future tasks, and suitably underbid in the current auction. Further, using feed-back from the auctions we adjust the weight, α placed on the long-term profits.

Formally, given a task t, drawn from a distribution $\mathcal{T}$, we define the marginal cost for state, S as $C_S\colon \texttt{Task} \to \mathbb{R}_{>0}$, where a state is the set of acquired tasks. Then, we submit the following bid,

$$\mathrm{Bid}_S(t) = (1-\alpha)C_S(t) - \beta\big(\mathop{\mathbb{E}}_{u\sim\mathcal{T}}[\ C_S(u)\ \ - C_{S\cup t}(u)]\ \big).$$

Here, $\alpha \in [-0.1, 0.3]$ and $\beta \in [0,1]$ are functions of the state, S and the history of auctions. They represent the risk-taking behaviour and the weight on long-run respectively. On receiving the auction results we update α and β as follows,

1. **(Reducing risk-taking behaviour).**
   For any $\alpha < 0$ we incur a loss in the long run. We gradually decrease α from 0.2 to -0.1, staring with under-bidding marginal cost, and eventually using the competitive advantage to bid above marginal costs. We increase α to adapt to the other-agents in the environment, thus taking a more aggressive strategy if we loose auctions. We change α at every auction as follows,

   $$\alpha(S) = \alpha_0 - 0.3\sigma(\frac{|W|-|L|}{2}),$$

   where, $|W|$ is the number of auctions won , $|L|$ is the number of auctions lost and $\sigma(\cdot)$ is the sigmoid function. α_0 is constant we tune in experiment 3.

2. **(Improving profits over time).** β is the weight we assign to future profits. If $\alpha = 0$, then for all β, our expected loss per task is $-\beta\big(\mathbb{E}_{u\sim\mathcal{T}}[\ C_S(u)\ \ - C_{S\cup t}(u)]\ \big)$. String with a "moderate" $\beta = 0.1$, we update it on every auction won. Specifically, after every auction we set,

   $$\beta = \beta_0\frac{0.9^{|W|}}{10}$$

   where, $|W|$ is the number of auctions won. β_0 is constant we tune in experiment 3.

2 Results

2.1 Experiment 1: Number of tasked auctioned

In this experiment, we test the performance of our agent against naive agent describes below.

2.1.1 Setting

We compare out agent against the `naive` agent described as follows,

1. `naive`: Completes tasks one after the other and bids the marginal cost.

For this experiment we use descriptions from `auction.xml` with seed= 123456. For this experiment we used the default time-outs.

2.1.2 Observations

Agents	**Tasks**= 5	**Tasks**= 10	**Tasks**= 15	**Tasks**= 20
`main`	812	734	1679	3123
`naive`	0	29	0	29

Table 1: Changing number of tasks, A

Agents	**Tasks**= 5	**Tasks**= 10	**Tasks**= 15	**Tasks**= 20
`main`	116	613	1432	321
`naive`	29	473	29	0

Table 2: Changing number of tasks, B

We observe that our agent gains an early competitive advantage and outperforms "dummy" on majority of the tournaments. In many cases we end up taking all of the available tasks. This corroborates our hypothesis that under-bidding early on is profitable in the long run.

2.2 Experiment 2: Profit per task

In this experiment, we test the performance of agents in terms of profit per task acquired. We test our agent against the naive agent by varying the number of tasks auctioned.

2.2.1 Setting

We compare out agent against the `naive` agent described in experiment 1. The specifications of the agents are same as experiment 1. For this experiment we use descriptions from `auction.xml` with seed= 123456. For this experiment we used the default time-outs.

2.2.2 Observations

Agents	**Tasks**= 5	**Tasks**= 10	**Tasks**= 15	**Tasks**= 20
`main`	162	92	112	173.5
`naive`	–	14.5	–	14.5

Table 3: Profit per task, A

The dummy agents achieve roughly equal profits per task in all setting, where as our agent doesn't display a consistent pattern. This suggests a possible of improvement on our agent. Specifically, by ensuring observing the bids of other agents we can increase our bids to ensure higher profits for

2.3 Experiment 3: Tuning hyper parameters

In this experiment, we test the performance against agent of group 17. We use our the results to tune the constants α_0 and β_0 described earlier.

Agents	**Tasks**$=5$	**Tasks**$=10$	**Tasks**$=15$	**Tasks**$=20$
`main`	39	68	110	16
`naive`	14.5	473	14.5	–

Table 4: Profit per task, B

2.3.1 Setting

We vary initial α and β, and compete against their agent. For this experiment we use 20 tasks in with descriptions from `auction.xml` and seed$=123456$. We set a time-out of $15sec$ per bid. As in one situation our agent is at a disadvantage, we use the results acting as a company A for hyper-parameter tuning.

2.3.2 Observations

α	$\beta = 0$	$\beta = 0.1$
0.0	1024	1524
0.10	1021	1498

Table 5: Hyper parameter tuning

We found that our agent performs the best with $\alpha = 0.0$ and $\beta_0 = 0.1$, i.e., with low risk taking behaviour and moderate weightage on future tasks. Since, β already contributes to under-bidding in the auction, we believe adding and additional risk may be detrimental. We would like to point out that this experiment is against a specific agent and with different strategies other hyper parameters may be optimal.