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

Group №45: Piquet, Erbacher

November 25, 2018

1 Bidding strategy

1.1 The bid's base

First, we wanted to determine to cost induced by the new task. To realize this calculation, we calculate an pseudo optimal plan with and without this task. Afterwards, we simply calculate the difference between the both costs. Of course, this difference won't represent the real cost induced by this task at the end but it gives us an idea of how far the task is from our current tasks.

1.2 The profit's calculation

Thanks to the calculation above, we approximately know the cost of the task. We thus need to add the profit. This number will depend of three things: the topology, how far the task is from the current tasks we won, and the previous bids of the other competitor.

1.2.1 Adaptation to the topology

Based on the average kilometers between the cities in the topology, we adapt a number which influences the profit. More the average kilometer is high, more we will increase the profit we want. For instance, the average kilometer in the french topology is very high, so the profit we will first try to make will be higher.

1.2.2 Adaptation to the distance between the new task and the others

Thanks to the bid's base calculation (section 1.1), we can determine how far the new task is from the others. More the bid's base is high, more this distance will be high too. If the task is far, we will bid a simple logic : base (section 1.1) + profit (section 1.2). However, we often have a task that has a very low cost. Plus, sometimes, a task induced a negative cost, that mean the plan without the task is costlier than with the task. This effect is mainly caused by the fact we don't build optimal plans but it means that the task is on our road. In this case, we can bid a very low cost to avoid the other competitor doing a profit on it.

1.3 Analyzing the other competitor's behavior

This section answers to: how do you use the feedback from the previous auctions to derive information about the other competitors?

1.3.1 Analysis

We wondered how can we find the other competitor's strategy. We choose to determine if he calculates his bids based on our bids. To verify that we send large bids on the second and third task and verify if he increased a lot his bids.
To verify it, we calculate two pseudo optimal plans, one with the tasks he previously won, another with these tasks plus the new task to bid. We thus determine a cost induced by the new task. We do that for the first three tasks and we consider that he was influenced by our large bids if he increases his bids by two.
We are doing large bids on the second and third tasks to get his average bid, without any influence, on two bids (the first one and the second one).
We also though that other competitor will want to do that with a large bid on the first task, we thus send a bid like: base (section 1.1) + 2* profit (section 1.2) to make better profit in this case.

1.3.2 Adaptation

If we detected that the other competitor is influence by our bids, we send a very large bid on the 8th tasks and increase our profit on the next 3 tasks by two.
If we think that the other competitor is not influenced by our bids, we calculate how many tasks we got. If we are getting less than the half of the tasks, we progressively decrease or profit every two tasks (by multiplying it by 0.92) until we start getting the task.
If we detect that we are winning more than the half of the tasks, we progressively increase our profit.

2 Results

In our code, we have 2 different strategies when the adversary take in to account our bids or not, we test our agent against two simple agents, one that adapt to our bids, and one that is blind.

2.1 Experiment 1: Comparisons with dummy agents

We compared our result with the dummy agent provided. We want to know what parameter makes our agent perform better.

2.1.1 Setting

On this experiment we want to see the influence of the ratio that is decreasing our bids when we don't get the last task. We ran many simulation between the dummy agent and our agent, with a variation of the ratio.

2.1.2 Observations

Our agent won all the rounds, with an average profit of 2057 against an average of 282 for the dummy agent. This is not a surprise, because the dummy agent makes random bids. But when the ratio is too low, our profit is really low either because the bids are too small really fast. And when the ratio is too high, the profit start to decrease as well. Maybe because the next bids has only too small variation and the agent still don't have the task. Even if there are lot of random in the results, we found a good balance between having the task and profit with a ratio around r = 0.8

2.2 Experiment 2: Comparisons with agent that takes bids into account

In this section the dummy agent tries to have a bid just under the average of our previous bids.

2.2.1 Setting

We did the same experiment but this time with an agent that took our bids into account. We made small changes on the ratio that influence our bids.

2.2.2 Observations

With an optimal ratio, our agent still won all the rounds, despite the fact that the dummy agent tries to converge under our bids. In our code, we intentionally bid very high at a specific moment in the auction, if we detect that the other agent tries to converge to our bid's average, we reduce our bids, with certain rate. When the rate is low, we get almost all the tasks but our profit are negative and we loose against the other agent. When the ratio is high, the other agent was fooled by our false bid for only 1 or 2 rounds, because we decrease our bids to slowly to stay under the next bid of the opponent.

Maybe we should have consider the distribution of the task in the topology.