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

Group №1: Célia Benquet - 271518, Artur Jesslen - 270642

November 22, 2020

## 1 Bidding strategy

Our main idea consists in calculating the marginal cost ($cost$(plan with extra task) $-$ $cost$(current plan)) for each new auctioned task and multiply it by a risk ratio that represents the strategy adopted to calculate the next bid price and a probability distribution ratio that depict how taking the extra task might be interesting given hypothetical future tasks.

The plans are computed using a local search optimization algorithm. It solves the problem as a discrete constraint optimization problem (COP) by finding the best plan for, on one hand, the obtained task and, on the other hand, the obtained tasks plus the auctioned one. Then, using the cost of these plans, we can easily find the marginal cost. We implemented our bidding strategy through the **BidStrategy** abstract class. We created subclasses for each form of strategy. Consequently, we defined **RiskyStrategy**, **CatchingupStrategy** and **SafeStrategy**. The resulting risk ratio vary depending on the strategy.

Our optimal strategy uses a combination of those three and starts with a risky strategy until the agent wins two tasks. It consists in bidding at a price lower than our marginal cost in order to have higher chances to win the first tasks. Indeed, it is easier to choose optimal tasks once we already have a set of attributed tasks. We also add a notion of how interesting the very first tasks are by looking at the probability that its **deliveryCity** and **pickupCity** are implicated in future tasks. However, the resulting utility is negative. Hence, the catching up strategy aims at winning back the utility that has been lost during the risky strategy. Therefore, the strategy always computes a bid higher than the marginal cost and tries to maximize the profit while it is winning tasks. Finally, once we get back to a neutral utility, the safe strategy consists in selecting the interesting tasks, with respect to the tasks already obtained, and choosing a bidding price in consequence. For that, we compare our utility to our opponent's utility. It gives us an information on how we perform. We also use the probability distribution of the tasks to estimate the potential future gain that the auctioned task could give us. For that, we have a distribution ratio between 0.9 and 1.1, used in the risky and safe strategy. It consists of the maximum of the probabilities between the probability that the **deliveryCity** is the **pickupCity** of a future task and the probabilities that the cities involved in the auctioned task is implicated in future tasks involving the cities from the tasks already obtained.

To derive information about the opponent, an estimation of how he is performing is considered. First, we estimate the initial cities for the vehicles of our opponent. However, we can only approximate the initial city of one vehicle and all other vehicles are given a random initial position as the first opponent bid gives us information to approximate the position of his closest vehicle, which would have handled the first auctioned task. For that, we select the city that minimizes the absolute difference between his bid and the bid we would have made from this city with our vehicles. Note that we hypothesize that the opponent has exactly the same vehicles and the same aggressive strategy as we have. It will, generally, in the worst scenario only overestimate its initial cost (if he is in fact not as aggressive). Then, during the risky strategy, we don't take into consideration our opponent to compute the risk ratio. However, for the catching up and safe phase, we also use the opponent bid and our estimation of its marginal cost to compute its potential risk ratio and we update it using a moving average.

# 2 Results

## 2.1 Experiment 1: Comparisons with dummy agents

### 2.1.1 Setting

We ran the tournaments for a dummy agent following a naive strategy against our best agent, which varies its strategy and takes into account the probability distribution of the task. Tournaments were run for 5, 10, 20 and 30 tasks.

### 2.1.2 Observations

We observe that for a low number of tasks, our best agent performs worst than the dummy agent. It makes sense as our optimal strategy consists in a bellow-cost bidding for the first 2 tasks in order to guarantee a fast first win. Hence, for 5 auctioned tasks, it does not get enough rounds to compensate for its initial losses. For 10 tasks and more, our best agent generally wins as the combination of strategies can really be used to its fullest.

## 2.2 Experiment 2: Comparisons of our different strategies

### 2.2.1 Setting

We ran the tournament with agents following different strategies: the risky, catching up and safe strategies as well as our best agent adapting its strategy depending on the stage of the game. After observing our results, we complemented our experiment by comparing our best agent to a strategy that doesn't switch to the safe strategy as final strategy but stay in catching up strategy. The tournament was run with 20 auctioned tasks.

### 2.2.2 Observations

Our best agent is performing the best. The risky strategy permits to its agent to get most of the tasks, however its utility ends up, most of the time, negative. In some cases, it manages to reorganize its obtained tasks in a way that the final utility is positive, which even makes it win the game in some rare situations.

Using a catching up strategy, the agent wants to maximize its reward and, so, it does not underbid. If it has a lower number of tasks, the tasks it got are worth a good amount of money. Hence, the agent performed nicely and even similarly to the best agent. It also wins over the agent following the safe strategy. Indeed, this last one relies on the utility of the opponent and makes sure that its own utility is higher but is more likely to take risks when its utility is positive and consequently bids lower.

To show that using a safe strategy is still useful for the best agent, we made it compete against an agent implementing our best strategy, only without switching to the safe strategy (only risky and catching up). Results showed that our best agent always performed better. It confirms that the safe strategy is an efficient strategy when it is combined with other strategies.

## 2.3 Experiment 3: Varying internal parameters

### 2.3.1 Setting

We ran the tournaments with different versions of our best agent. We changed the way we update the risk ratio by changing the **epsilon** constant to 0.05, 0.1, 0.2 and 0.5 (our default value is 0.1). It corresponds to changing the speed to which an agent can adapt during the game. We also compared our best agent with and without considering the probability distribution of the task with a distribution ratio bounded between 0.9 and 1.1 and 0.8 and 1.2.

### 2.3.2 Observations

Regarding the risk ratio calculation, we observed no significant difference in performances for an **epsilon** bellow 0.2. However, up this value, the performances decrease as the incremental variation of the risk ratio seems to be too big. Indeed, the agent fits its risk ratio too much to the final results of the previous round, making it very specific to the last auctioned task and less adaptable to answer to the next round.

Comparing our best agent with and without taking the probability distribution of the tasks shows that a strategy that takes it into account performs slightly better. It makes sense as it considers the future and not just the previous bids and the current marginal costs. However, by making the distribution ratio able to vary more (0.8 to 1.2), the results are worst. Indeed, varying the bid by extensively considering the future results in decreasing the weight that surer information, such as the previous bids and current marginal cost, has for the decision of the final bid. We found that a distribution ratio between 0.9 and 1.1 is a nice compromise.