An Auctioning Agent for the Pickup and Delivery Problem

This exercise is graded with 100 points.

Centralized coordination is guaranteed to produce the optimal plan (globally or locally) for a multi-agent PDP problem, when :

— All tasks are known in advance (they were until now, in our setting)

— The company has complete information about the parameters of its vehicles

— The vehicles blindly follow the orders of the company

Unfortunately, these conditions are not always met in real life applications. Most often vehicles or group of vehicles (i.e. companies) are self-interested and might not be willing to obey a central planner. This may happen for various reasons : for example, the central planning might be unfair, or the agent might not wish to reveal the true information about its state.

One intuitive solution is to allow the agents to negotiate and distribute the transportation tasks among themselves, such that they coordinate their actions in a decentralized fashion. A market is thus created, where the tasks are "sold" to the agent that is most willing to take them. The competition usually leads to an efficient delivery solution.

Competing for the best strategy !

In this exercise you will learn to design agents that compete against each other within the framework of a package delivery problem. Each group will be asked to design an agent that will be competing against each other. The goal of each agent is to maximize its profit, i.e. to maximize the difference between the reward for delivering all tasks and the transportation costs.

We will use a *closed-bid first-price reverse auction* to allocate the tasks to the agents. We assume that there is a trusted auction house that is responsible for auctioning the set of tasks. At the beginning of the simulation, there is an auction house that auctions the tasks one after another to the agents.

For each task to following steps are taken :

1. The auction house publishes the details of the task, i.e. the pick-up city, the delivery city, and the weight of the task are revealed to the agents.
2. Every interested agents may then submit bids for the task. The bids are integer numbers, representing the payment that the agent requests for the delivery of the task.

3. The task is assigned to the agent with the lowest bid and that agent is paid according to his bid.

Consider the following example : A task T1 has to be transported from Lausanne to Geneva. Assume that agent A bids 480 CHF for the task and agent B bids 440 CHF. Agent B has the lowest bid and wins the task. This means that agent B has to deliver the task T1 and it will be paid 440 CHF for it. The actual profit that B makes from this task will depend on the efficiency of its plan.

You should note that once an agent has "won" a task it is responsible for its delivery. Agents cannot later refuse to deliver the tasks they have won in previous auctions. Another thing to bear in mind is that the total payment received by the an agent (the sum of all winning bids) does not depend on the route traveled by the agent. It is therefore in the best interest of every agent to compute a plan that minimizes the total cost for delivering the assigned tasks.

Your task

You will define an auction strategy for your agent and compete against other agents. When designing your strategy, please keep in mind the following facts :

— The minimum price you are willing to accept for delivering the task T1 is equal to the marginal cost of delivering T1. In other words, you have to (1) find the cost of the plan for delivering the tasks you've already won, (2) estimate the cost of the plan for delivering the same tasks plus T1, and (3) compute the marginal cost of delivering T1 as the difference between the two.

— Any price paid by the auction house which is superior to the marginal cost of a task will result in a profit for you to deliver that task. Therefore, you should usually bid above your own marginal cost. You may occasionally bid below your marginal cost if you believe this will reduce the cost of future tasks, but be careful not to end up with a deficit !

— You do not know what other tasks will be auctioned by the auction house. In fact, you do not even know **how many** more tasks will be auctioned after the current one. When you compute your current bid, you might want to speculate on the effect of taking the present task on the future auctions.

— You do know the probability distribution of tasks. The tasks that will be auctioned will follow this distribution, so you might want to use these probabilities in order to speculate about future tasks that will be auctioned.

— The winner and all bids are published immediately after each auction. This gives you the possibility to refine your strategy depending on your opponent(s).

Feel free to try competing between each other to improve your agents. You can even create your own tournaments using the command line arguments of the *LogistPlatform*.

Competition rules

After you submit your agents we will run a competition with the the following rules :

- Every group is entitled to subscribe one agent to the competition.
- Every agent is represented by a company controlling between 2 and 5 vehicles.
- We (the organizers) may add any number of default (dummy) agents whose strategies are kept secret.
- The competition is run on several configurations. The configurations are fair but need not be symmetric for all companies. For instance, company A could have 2 medium vehicles while company B has 1 small and 1 large vehicle.
- In order to provide ultimate fairness the competition on each configuration is repeated several times for different permutations of agents. In the example above your agent gets to play both the role of company A and the role of company B in some run.
- Each task is assigned to the lowest bidder, ties are broken using the company id. The final plan of each agent must handle all tasks that the agent has won. Agents who submit an incorrect plan lose the current game. Agents that timeout will also lose the current game.
- The winner of the competition is the agent with the largest number of won games.

Implementation Hints

- You will need to implement the `AuctionBehavior` interface. The interface is slightly more complex than usual so you may want to have a look at the API or the *LogistPlatform* documentation.
- You can probably reuse parts of the previous project in order to compute an efficient plan.
- Focus on refining your strategy rather than optimizing your planner. A good strategy can have a larger impact.
- Make sure that your agent obeys the time limits given by the platform.
- Again, if the *LogistPlatform* detects a problem in your plan, the simulation will exit and a detailed error message is displayed. Remember that you have to handle all tasks and that each tasks can only be picked up and delivered by one vehicle.

Deliverables (Due on Sunday 24.11.2019, 23 :55) :

What to submit !

— A report in PDF describing the strategy of your agent in detail, evaluation of the algorithm and your findings
The report should be based on a latex template that is given on the exercise description moodle page.
— Your source code and compiled code (**JAR file**), implementing your bidding strategy
— the name of your agent (in `agents.xml`) that represents your final solution should be `auction-main-[group number]`, for example, `auction-main-01` or `auction-main-40`
— **NB! The agent `auction-main-[group number]` must be the only agent in the config and in the JAR file.**
— rename the auction folder to lastname1-lastname2-ex5
— create a directory called doc within this folder and place the report in this folder
— create a zipfile lastname1-lastname2-ex5.zip containing everything in the lastname1-lastname2-ex5 folder