Auctioning Agent

Jeremy Gotteland & Quentin Praz

December 3, 2015

Improvement of the centralized agent

In this exercice, we chose to use the centralized agent that we implemented before. In order to have better results, we decided to improve it first.

The first improvement was to stop the oscillation of the agent between two equivalent solutions. Indeed, we realize that some solutions were equivalent and that our agent was stuck in these solutions. For exemple two equivalent solutions are:

$$S1 : \{..., Pickup(c1, c2), Deliver(c0, c1), ...\}$$

$$S2 : \{..., Deliver(c0, c1), Pickup(c1, c2), ...\}$$

If the vehicle has enough space to pickup then deliver, these two solutions are equivalent. Then, we add a condition in our validation method that states that when we have a pickup task and and delivery task in the same city, we always deliver first.

The second improvement was to change the stochastic local search. The new method we chose is the following:

- With a probability p, we select the best neighbor.

- With a probability $p - 1$, we select a random neighbor.

And the last improvement was to keep track of the best solution we saw, even if it's not the last one, and to return it.

Implementing a bidding strategy

For every task available, agents have to bid following a *closed-bid first-price reverse auction*

Basic agent

Our basic agent's bidding strategy is pretty simple:
First we compute the marginal cost, which is the difference of cost between a plan with the requested task (computed via previous homework's SLS algorithm) and the current plan we have.
This marginal cost is the minimum amount we can bid, because if we bid less we will lose money (get rewarded less than what it actually cost us to deliver the task). So we tweak that cost to make a little profit, while trying to stay under the radar of other algorithms trying to guess what our bidding strategy is:

If we call C_m our marginal cost, and p is a random number between 0 (included) and 1 (excluded) then the formula for our bid B is:

$$B = (1 + 0.5.p) * \begin{cases} 7000 \text{ if } p > 0.9, C_m > 1000 \\ 50 \text{ if } C_m = 0 \\ C_m \text{ otherwise} \end{cases} \quad (1)$$

The Hollywood (Star) Agent

The Hollywood agent has a similar bidding strategy, except it also takes into account **opportunities** brought by getting the task. We define the opportunity score of a task to pickup in city A and deliver to city B by:

$$O_{AB} = \frac{\sum_{C=A+1}^{B} P_t(C, B)}{nbCities(A, B) - 1} \quad (2)$$

Where $P_t(C, B)$ is the probability of having a task from C to B, $nbCities(A, B)$ the number of cities between A and B, and $A + 1$ being the city right after A on the shortest path from A to B.

This opportunity score can help us bid less for promising task that will be likely to take us in a very comfortable situation later on. The equation we get for the bid is now:

$$B = (1 + p - O_{task.pickup,task.delivery}) * \begin{cases} 7000 \text{ if } p > 0.9, C_m > 1000 \\ 50 \text{ if } C_m = 0 \\ C_m \text{ otherwise} \end{cases} \quad (3)$$