Reactive agents

Jeremy Gotteland & Quentin Praz

October 14, 2015

World representation

State representation

Recall that the agent here refers to a single vehicle. We chose to represent the state of an agent as the following:

- Current city: The city in which the agent is currently located.
- Destination city: At any time the agent is a state where it could pick up a task to bring to a city. This variable is this potential city. If there is no task to be pick up, this variable is set to $NULL$.
- Possible moves: A list of cities which the agent can travel to from current state i.e. its neighbor cities and its potential destination city.

Actions

In our state representation, an action is just a move to another city.

Reward

The reward for moving from a state to another city c is defined as:

$$R(s, c) = \mathbb{1}_{\{c\ =\ s.destinationCity\}} \cdot AR(s.currentCity, c) - Km(s.currentCity, c) \cdot CostPerKm$$

Probability of transition

The probability of transition to the state s' from the state s by moving to city c is defined as:

$$T(s, c, s') = \begin{cases} P(c, s'.destinationCity) \text{ if } destinationCity \text{ is not } NULL \\ 1 - \sum_{c' \in Cities} P(c, c') \text{ otherwise} \end{cases}$$

Algorithm

Here we describe the algorithm we used

Implementation details

Observations

The only parameter we could play on was the discount factor. We tried some very small values and some values close to 1 but we didn't notice major changes. The only thing that was modified was the time of the pre-computation: the convergence was faster with a small discount factor.
One reason could be that networks are quite small and then the futur events don't count that much in the equation.