Pickup and Delivery Reactive Agent: Implementation of MDP

Christophe Marciot & Titouan Renard

September 30, 2020

1 Problem definition

1.1 Markov Decision Process

In a MDP current state is know with certainty, but the reward of transition is not. A MDP is defined by :

$$\textit{A reward function:} \qquad \overbrace{R(s,a) \to \mathbb{R}}^{\textit{Where } s \textit{ denotes a state and } a \textit{ an action}}$$

$$\textit{A probabilistic state transition table:} \qquad \overbrace{T(s,a,s') = p(s'|s,a)}^{\textit{Where } s' \textit{ denotes the state the action leads to}}$$

The goal of the process is to find a policy π such that *the average reward is maximized*.

1.2 The Pickup and Delivery Problem

Agents exist in a static environment (a model of Switzerland's road network) described by a graph. Nodes of the graph are called *cities* and it's (weighted) edges are called *roads*.

The pickup and delivery problem is described by a series of tasks spread over the topology, the transportation tasks are described by:

1. Pickup city (and it's position)
2. Delivery city (and it's position)
3. Reward in CHF

1.3 Definitions

1.3.1 State

It doesn't seem to make sense for the state to be anything other than **the city the agent is when it has no task**, this is because :

1. An agent can not have more than one task
2. When an agent has a task it cannot interrupt it
3. There is no difference between an agent in a city with no task because it succeeded or because it failed it's last task, it still has to make a decision about how to get another task

The set S containing all states is exactly the set of all cities.

1.3.2 Action

An action consists in the agent either:

1. going to a city with the goal of finding and completing a task there
2. taking a task in the city it's in

And always result in the agent being in a city (different or not from it's starting point, the agent can loop between two city if it maximizes reward) without a task, in another words in a (new) state.

The set A containing all actions is ...

1.3.3 Reward

Where $i(a)$ is the starting city of a given task corresponding to a given action, $j(a)$ the city it ends in and $t(a)$ the time it takes to complete the task in case of a success.

$$R(s, a) = \frac{R(i(a), j(a))}{t(a)}$$

1.3.4 Probability of transition $p(s'|s, a)$

1.3.5 Existing tables

The dataset usable for learning is described by two probability tables :

1. $P_{table}(i, j)$: the probability of a task for city j to be present in city i
2. $R_{table}(i, j)$: the average reward given when a task is transported from city i to city j

2 Solving the MDP

We denote *the value of a state* s as $V(s)$. This value represents "*the potential rewards from this state onwards*". In order to ensure $V(s_i) < \infty \; \forall i$ (and make the problem solvable) we introduce a *discount factor* $\gamma \in [0...1[$.

$$V(s_i) = R(s_i) + \gamma \cdot V(T(s_i), a(s_i))$$