Excercise 3
Implementing a deliberative Agent

Group №28: Mateusz Paluchowski, Vincent Smet

October 24, 2017

1 Model Description

1.1 Intermediate States

Every state is represented as follows:

State = (Initial City, Current City, Tasks in the trunk, Tasks to be picked up, Plan)

1. Initial City: The city where the vehicle starts. This initial city attribute is used to be able to perform a copy of a state.
2. Current City: The city where the vehicle is at this moment.
3. Tasks in the trunk: A list of tasks that already have been picked up by the vehicle and still need to be delivered.
4. Tasks to be picked up: A lists of tasks that still need to be picked up.
5. Plan: The plan to get from the initial state to the current one.

In effect the initial city and plan store the history of actions made by a vehicle and the current city and task lists describe the current state and provide the information needed to derive the possible successor states.

1.2 Goal State

The goal state is a state where both the Tasks in the trunk and Tasks to be picked up are empty. In such a state all packages have been delivered.

1.3 Actions

Possible actions are picking up a task of Tasks to be Picked up or delivering a task of Tasks in the trunk. Performing this action vehicles choose the shortest way to the pick up- and delivery city respectively. Doing so the vehicle will also perform all possible pick-ups and deliveries in the cities it crosses.

2 Implementation

2.1 BFS

Following the standard BFS algorithm, different layers of the search tree get constructed through a successor function that uses the actions described above. As a consequence the solution that is found first by breadth first will be the one that combines as much pick up and delivery actions in one step on average.

2.2 A*

For the Astar algorithm we sort constructed successors according to the reward resulting from the actions taken before adding them to the search tree. This way when two solutions are on the same level of the tree, the one with the most rewarding moves will be chosen.

2.3 Heuristic Function

$$Reward = -CostPerKM \times distance + \sum taskspickedup.reward \times 100$$

The heuristic favors picking up packages over delivering them. (Note that the agent only picks up when he has the capacity to do so) In consequence the agent will try to pick up packages as much as possible as long as he has the capacity and as long as there are packages to be picked up. From the moment the capacity is reached or in an absence of packages to be picked up it will start delivering packages as close to his current location as possible.

3 Results

3.1 Experiment 1: BFS and A* Comparison

3.1.1 Setting

First off, to compare performance a selection of different number of tasks generated was made. Run times and reward per kilometer were compared.
Secondly, for each algorithm separately the number of tasks generated was increased up until the point that it took more than one minute to generate a plan.
Both experiments were performed with a single vehicle in the Switzerland topology.

3.1.2 Observations

Reward per KM Number of tasks	BFS	ASTAR
6	2.48	2.48
7	2.115	2.720
8	3.073	3.885
9	2.630	3.325

Table 1: Optimality of BFS vs. ASTAR

From the table above it can be concluded that ASTAR becomes increasingly more optimal than BFS for larger problems. The plans to be implemented become longer and the difference in optimality becomes more significant.

Time[s] Number of tasks	BFS	ASTAR
6	0.037	0.019
7	0.219	0.058
8	1.143	0.201
9	52.39	9.664

Table 2: Efficiency of BFS vs. ASTAR

The same trend is to be seen in efficiency of calculating the plans for more complex problems.

	nr. tasks	time[s]
Astar	12	56
BFS	9	51

Table 3: Number of tasks treatable under 1 minute

3.2 Experiment 2: Multi-agent Experiments

3.2.1 Setting

Multi-agent experiments consists of various number of agents raging from 2 to 5.
In order to make calculations feasible we settled on default number of tasks for both BFS and A-Star algorithm equal to 6. Moreover we performed the experiment on default Switzerland topology. Seed of 123456 was used for reproducibility.

	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Avg. Reward of Vehicles
experiment 1	140	215	-	-	-	177.5
experiment 2	55	215	110	-	-	126.67
experiment 3	55	215	110	-5	-	93.75
experiment 4	40	345	-5	-5	170	109

Table 4: Multi-agent rewards comparison for BFS algorithm

	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Avg. Reward of Vehicles
experiment 1	25	270	-	-	-	147.5
experiment 2	25	170	90	-	-	95
experiment 3	25	215	-5	145	-	95
experiment 4	55	105	-5	-5	195	69

Table 5: Multi-agent rewards comparison for A-Star algorithm

3.2.2 Observations

One can clearly see that multiple agents while performing their plans accordingly to both BFS and A-Star solutions are in fact competing against one another, thus resulting in suboptimal final solution. Some agents can be extremely unlucky in situation where there are acting in accordance to policy 'shared' by other agent - basically mimicking steps however arriving at the pickup too late and not being able to get any tasks. The trend where more agents means more competing and thus less optimal general solution seems to hold, with one exception for BFS with 5 agents where there was unexpectedly high reward for agent 5 resulting in somewhat high average reward, however we treat this case as an outlier.