Autonomous Systems LAB Session 1: Search

Arnau Colom, Emma Fraxanet

I. FINDING ALL THE CORNERS (*Q5*). STATE REPRESENTATION.

The state of the corners Problem can be defined only by the current position of Pacman (x,y) and a list of the visited corners. Each time we search for the successors of the current position of Pacman we will check if one of the target corners has been visited and update the List of `visited_corners`. To know if all corners have been visited we do a simple conditional. If the number of visited corners is the same as the number of corners that we had as targets at the beginning, we consider the problem solved.

II. CORNERS PROBLEM (*Q6*). HEURISTIC

To obtain a good heuristic what we propose is estimating the distance it would take to visit all non-visited corners successively. Since the cost is uniform this distance translates into the estimated cost. This is done by estimating the distance to all non-visited corners from the start state and picking the shortest distance (greedy choice). Assuming we would move to that corner first we eliminate that corner from the non-visited corners and do the whole procedure all over again. Finally we sum up all obtained minimum distances, obtaining an estimation of the shortest path that goes through all non-visited corners. The returned value is then not overestimating the distance, its estimation is always lower or equal to the true one and thus it's admissible.

In this case we can sense the consistency of the heuristic because we are using the Manhattan distance. Since our system is a grid that allows 4-direction actions, each with a uniform cost, moving to a new successor doesn't drop more than the uniform cost in the heuristic. It is hard to prove but the fact that we consider the sum of all distances doesn't allow a big change in the heuristic cost when moving only one position in the grid.

III. FIND ALL FOODS(*Q7*). HEURISTIC

The intuition followed to compute the heuristic for the 'Eating All The Dots' problems is strongly based on the resolution of Q6. To compute the heuristic in this problem, we estimate the distance it would take to visit all the positions of the Food-Grid that represent the food points, i.e., the positions of the grid that contains `True`.

In this case we change the distance used, instead of applying the Manhattan distance we used the `mazeDistance()` function provided by the Project. This distance takes into account the walls in that system and therefore provides a better estimation. Another difference in this heuristic is that the total distance is computed adding the distance to the furthest food (the maximum instead of the minimum). Then we again consider that we move to that position and repeat the procedure. Finally this total distance is divided between the number of foods to obtain the average. The result would be the average path that Pacman would take to go from a certain position to the further piece of food. Assuming it is possible to pick up all other foods before reaching that one, this heuristic provides a good estimation (doesn't overestimate the distance). Our intuition tells us that the fact that we compute an average helps provide consistency to the problem.