Özgur Cem Birler
50118

Written Q1: What are some differences between DFS and BFS in terms of path cost and number of expanded nodes? Which one would you pick when and why?

DFS expands the deepest unexpanded node whereas BFS expands the shallowest expanded node.

In bigMaze DFS and BFS yields the same 210 path cost, DFS expands 390 nodes whereas BFS expands 620 nodes. In openMaze DFS gives 212 path cost with and expands 576 nodes whereas BFS gives 54 path cost and expands 682 nodes. If a graph doesn't have too many branches, then DFS makes more sense since it wont expand the neighbors unless last node explored doesn't have any more successor and the goal is not attained.

In general, I would choose DFS over BFS, assuming the problem doesn't have a node with infinite depth, because even though DFS expands more nodes it requires significantly less amount of space: O(b*m) instead of O(b^d). Since space can grow larger exponentially, for even basic problems we can get into trouble with BFS.

Written Q2: What are some differences between UCS and A* in terms of path cost and number of expanded nodes? Which one would you pick when and why?

UCS expands the node with the lowest path cost whereas A* expands the node with the minimum lowest path cost + heuristic cost.

In bigMaze both UCS and A* search yields 210 path cost, UCS expands 620 nodes whereas A* search expands 549 nodes with Manhattan heuristic. In openMaze both UCS and A* search yields 54 path cost, UCS expands 682 nodes whereas A* search expands 535 nodes with Manhattan heuristic.

I would choose A* search since for both mazes, both have the same cost but A* search expands less nodes.

Written Q3: Comment on your choice of state in the four corners problem. Why does it allow you to solve the problem?

I chose the starting point and the coordinates of the four corners as the state in the four corners problem. So state is a tuple with 5 elements: [starting point, corner1, corner2, corner3, corner4] When Pacman arrives one of the corners that corner is updated with 0. If all the corners are updated to 0s then the state is the goal state. It allows me solve the problem because all we need to know is whether Pacman arrived all of the four corners. In order to do that we need to store information about each corner as well as the current position of Pacman.

Written Q4: Comment on your choice of heuristic in the four corners problem. Why did you settle on that heuristic? Why is it admissible and consistent?

My corners heuristic gives the maximum Manhattan distance to the undiscovered corners from the current position of Pacman. The heuristic is admissible because we know that in order to satisfy the goal the Pacman must arrive to every corner hence it must go to the furthest not discovered corner from its position which has always a lower cost. This heuristic is also consistent because it holds $h(n) \leq c(n, a, n') + h(n')$

which can be rewritten as: $h(n)-h(n') \leq c(n, a, n')$ (2)

If we assume no corners are discovered between states, then the difference between heuristics of state n and its successor n' is the Manhattan distance between Pacman's position in these states:

abs[h(n)-h(n')]=Manhattan distance(n,n') and this is always smaller or equal to cost of reaching n' from n which satisfies (2)

If a corner is discovered between states, then it it sufficient to consider the cases where the heuristic of the n is more than n'. Since otherwise left side of (2) yields negative. In the other case the furthest point doesn't change the value of heuristic doesn't increase so we can consider this as a point and the position at the stage n and position at stage n'. Now we if we apply triangle inequality we see that the (1) holds.

Written Q5: Comment on your choice of heuristic in the eating all the dots problem. Why did you settle on that heuristic? Why is it admissible and consistent?

My food heuristic is the maze distance between current position of Pacman and the closest food + number of foods left - 1. I settled on that heuristic because if the goal is to eat all of the dots, it makes sense so start with the closest one. This heuristic is admissible because if there is only 1 food left then Pacman must go to this food and the minimum cost of reaching that distance is given by maze distance. If there are more than 1 food left, then they must be in other positions, which the cost between two states is at least number of foods left. This heuristic is consistent because if the closest food doesn't change then Pacman would be at most 1 step closer to the closest food, if the closest food changes between states n to n' then cost of going to that state + cost of reaching to goal from n' is more than cost of reaching from n directly because Pacman needs to go previously closest food to eat all of the foods.

Written Q6: What are some practical differences between a consistent and an inadmissible heuristic, in terms of path cost and number of expanded nodes? Which one would you pick when and why?

A heuristic is admissible if it never overestimates the path cost, and it is consistent if reaching the goal from a successor state + cost of reaching to successor is always greater or equal to reaching the goal from the current state. Every consistent heuristic is admissible since even if we choose a successor (n') which is on the optimal path, $c(n, a, n') + h(n')$ will be equal to the cost of reaching the goal from n, which is always greater than the heuristic by the definition hence the admissibility follows. I would pick consistency over admissibility because it is more restrictive.