4511W – Homework #2

Kevin Dunn

dunn0407@umn.edu

E = explored

F = fringe

Node = (name, A* cost)

1. (1) E ={null}
   F={ (5,14) }

   (2) E = {(5,14)}
   F = { (5,16), (5,19) }

   (3) E = { (5,14), (8, 16) }
   F ={ (7,19), (1,19), (2,19) } // pick lesser node when equal costs

   (4) E = { (5,14), (8,16), (1,19) }
   F = { (7,19), (2,19), (9,24), (6,25) }

   (5) E = { (5,14), (8,16), (1,19), (2,19) }
   F = { (7,19), (9,24), (6,25), (4,20), (3,23) }

   (6) E = { (5,14), (8,16), (1,19), (2,19), (7,19) }

   F = { (8,19), (4,20), (3,23) , (9,24), (6,25) }

   (7) E = { (5,14), (8,16), (1,19), (2,19), (7,19), (8,19) }

   F = { (1,18), (2,18), (8,19), (4,20), (3,23) , (9,24), (6,25) } // pick lesser node when equal costs

   (8) E = { (5,14), (8,16), (1,19), (2,19), (7,19), (8,19), (1,18) }

   F = { (2,18), (8,19), (2,19), (4,20), (3,23) , (9,23), (6,24), (9,24), (6,25) }

   (9) E = { (5,14), (8,16), (1,19), (2,19), (7,19), (8,19), (1,18), (2,18) }

   F = { (4,19), (8,19), (2,19), (4,20), (3,22), (3,23) , (9,23), (6,24), (9,24), (6,25) }

   (10) E = { (5,14), (8,16), (1,19), (2,19), (7,19), (8,19), (1,18), (2,18), (2,19) } // pick lesser node when equal costs

   F = { (4,19), (8,19), (2,19), (4,20), (3,22), (3,23) , (9,23), (4,24), (6,24), (9,24), (6,25), (3,26) }

   (11) E = { (5,14), (8,16), (1,19), (2,19), (7,19), (8,19), (1,18), (2,18), (2,19), (4,19) }

Solution: 5 ->7->8->2->4 Cost: 19

2.

(1) The heuristic in problem 1 is not admissible. Node 9 has a heuristic value of 10, but a path exists {9 -> 1 -> 2 -> 4} that has a cost of 9. Since the heuristic is not admissible, then it is also not consistent.

(2) h_3 must be admissible, but not necessarily consistent. Suppose there are 3 nodes, A, B and Goal. $h_1(A) = 5$, $h_2(A) = 6$ and the optimal path from A to the goal is 100. The cost of going from A to B is 2. $h_1(B) = 3$, $h_2(B) = 1$ and the optimal path from B to the goal is 98. Both heuristics are admissible since their values are smaller than the optimal path to the goal at each point which makes them admissible. $h_1(B)$ + $\text{cost}(\text{A}, \text{B}) = 7 \geq h_1(A) = 5$ which is consistent. $h_2(B)$ + $\text{cost}(\text{A}, \text{B}) = 5 \leq h_2(A) = 6$ which makes it inconsistent. h_3 takes the min of either which means that it would begin with $h_1(A)$ and then deviate to $h_2(B)$ making it admissible, but not consistent.

3.

(1) A relaxation could be that your children are well behaved, and you can take any child at any time without the risk of fighting. The general optimal solution would be to take the children in any order to school one by one. The heuristic would be to calculate the number of children still waiting to be taken to school.

(2) A relaxation of the loop-puzzle might be that you do not have to connect all lines, but only required to make the appropriate number of lines around each number on the board. A general solution would be to make the appropriate number of fences around each number without concern about connecting them. The heuristic would be the total of the numbers on the board.

4.

(1) I think a hill climbing algorithm would work. Using the most cost-efficient weapons first to see if you can hit with their smaller radii and then make decisions based on how close the weapon hit to the target to determine if that is a good enough weapon that you can hit it or if you need to use a weapon with a larger blast radius.

(2) I would use would be a greedy first search. The reasoning behind this is having a good heuristic (Geiger counter ticks) and not having any specific movement restrictions I could simplify my searching by following the Geiger counter readings.

(3) For this maze I would use a BFS because it's guaranteed to find the end and find it with the shortest path and we don't have a heuristic to see if we're getting closer to the exit of the maze.

(4) I would use a uniform cost search by putting home as a start node, destination as an end node and gas station choices as nodes in-between. The distance and cost per gallon between stations is known, the branching factor for each gas station is low, and there is a definite goal state of maximum depth it would be feasible to iteratively search the lowest cost path.

5.

N	ID-DFS	DFS	GENETIC
11	12.480s	0.036s	0.363s
20		14.604s	1.080s
40			4.071s

ID-DFS: Iterative deepening can be useful in small instances of the n-queens problem as it only needs to iterate as far as a solution can be found. The memory is relatively small being O(bd), but since it must restart searches every iteration, longer solutions where n is large may take a long time.

DFS: Depth first search is useful for pinpointing a solution. It doesn't waste time exploring all nodes at a level, but that means that it has the chance of being unlucky and following down a bad path where a solution might not be possible. Since it explores at the end of a bad path it may waste time.

GENETIC: The genetic algorithm looks like a faster solution compared to the previous algorithms. It relies on probability to find its solutions, but that comes with risk that it may not find a solution in appropriate time. The other problem is that it was difficult to set up as it takes much more preparation and complexity to properly run.