CS 571 : Artificial Intelligence Lab

Sahil Gupta
1901CS49
Mayank Raj
1901CS35
Manjul Bamrara
1901CS32

1. Let this be a randomly generated grid ((6, 1, 7), (2, 0, 8), (3, 4, 5)),

the number of states explored by different heuristics:

h1: No of states explored = 152742 h2: = 32692

h3: = 1233 h4: = 5721

[[In terms of efficiency h1<h2<h4<h3]]

We can see in the above example that the number of states explored is in the order h1>h2>h4>h3. Since the h1 heuristic returns 0 for all the nodes, we can see that it's the worst in terms of efficiency and hence expands to a larger number of states. h2 which uses the count of the number of misplaced tiles as the heuristic expands the lesser number of states in comparison to h1 but more number of states in comparison to h3 (manhattan distance) and h4 (euclidean distance).

2. For this starting state: ((1, 2, 3), (4, 5, 6), (0, 7, 8))

Heuristic 4 expands the following states: ((1, 2, 3), (4, 5, 6), (0, 7, 8)) ((1, 2, 3), (4, 5, 6), (7, 0, 8))
Heuristic 3 expands the following states: ((1, 2, 3), (4, 5, 6), (0, 7, 8)) ((1, 2, 3), (4, 5, 6), (7, 0, 8))
Heuristic 2 expands the following states: ((1, 2, 3), (4, 5, 6), (0, 7, 8)) ((1, 2, 3), (4, 5, 6), (7, 0, 8))
Heuristic 1 expands the following states: ((1, 2, 3), (4, 5, 6), (0, 7, 8)) ((1, 2, 3), (0, 5, 6), (4, 7, 8)) ((1, 2, 3), (4, 5, 6), (7, 0, 8)) ((0, 2, 3), (1, 5, 6), (4, 7, 8)) ((1, 2, 3), (4, 0, 6), (7, 5, 8))

From the previous question, we know that heuristic 1 is the worst. We can see in the above example, that the states ((1, 2, 3), (4, 5, 6), (0, 7, 8)), ((1, 2, 3), (4, 5, 6), (7, 0, 8)) expanded by h2,h3 and h4 was also expanded by h1.

3.

Monotone restriction is h(n) <= cost(n,m) + h(m)

a. h1(n): return 0

b. h2(n) = number of tiles displaced from their destined position.

c. h3(n) = sum of Manhattan distance of each tile from the goal position.

a. h1(n) = 0
   h1(m)=0
   0<=cost(n,m)+h(m)
   If the states are adjacent i.e. we go to m just after expanding n. cost(n,m)=1
   Hence, 0<=1+0
b. h2(n) = number of misplaced tiles.
   Let's say we have a state n= ((1,2,3),(4,5,6),(0,7,8).
   From n we go to state m = ((1,2,3),(4,5,6),(7,0,8)) by incurring a cost of 1 unit.
   h(n)=3 and h(m)=2
   It can also be easily seen that for two neighboring states if h2(n)=c, h2(m) will either have one less misplaced tile(c-1) or one more misplaced tile(c+1) or the same value as h2(n)(c). cost(n,m) will be one for 8 puzzle problems for every change of state. Therefore our inequality will always hold.

c. h3(n) = sum of the Manhattan distance of each tile from the goal position.
   Whenever we go from state n to state m, the Manhattan distance will either increase by 1 or decrease by 1 i.e., if h(n)=c, h(m)=c+1/c-1. Also, the cost incurred on going from state n to state m is 1. Hence, our monotonicity holds.
d. h4(n) = sum of the euclidean distance of each tile from the goal position.
   Whenever we go from state n to state m, the Manhattan distance will either increase or decrease by a value 'v' which is less than or equal to 1 i.e., if h(n)=c, h(m)=c+v/c-v. Also, the cost incurred on going from state n to state m is 1. Hence, our monotonicity h(n) <= cost(n,m) + h(m) holds.

4.

We can see that for a randomly generated grid, ((1, 3, 6), (0, 4, 7), (5, 8, 2)) the solution doesn't exist. The algorithm explores all 191340 states and fails to reach the target grid.

Heuristic 4
Failed to reach the target grid!
No of states explored = 191340

Heuristic 3
Failed to reach the target grid!
No of states explored = 191340

Heuristic 2
Failed to reach the target grid!
No of states explored = 191340

Heuristic 1
Failed to reach the target grid!
No of states explored = 191340

5.

Monotone restriction: h(n) <= cost(n,m) + h(m)

a. h2(n) = number of tiles displaced from their destined position.

b. h3(n) = sum of Manhattan distance of each tile from the goal position.

6.

Suppose we add the cost of the empty tile in the heuristic. In that case, when we traverse from state n to state m, we incur a cost(n,m)=1 which is the same as h2 but, since we are counting the blank space in the number of misplaced tiles so after state change, the number of misplaced tiles can decrease by 2 in case state m is the target grid. Let's say state n = ((1,2,3),(4,5,6),(7,0,8)) and m = ((1,2,3),(4,5,6),(7,8,0)). In this case h(n) = 2 and h(m) = 0.

Also, the cost incurred is 1. Therefore, h(m)+cost(n,m) = 0+1 which is less than h(n). Hence, the monotonicity h(n) <= cost(n,m) + h(m) is violated.

4.

a. Total number of states explored.
b. Total number of states on optimal path.
c. Optimal path.
d. Optimal Cost of the path.
e. Total time taken for execution

Start state: ((1,8,7),(4,5,3),(0,2,6))

Heuristic	# of states explored	# of states on optimal path	Cost of optimal path	Total time for execution
h1	41301	21	20	1.486s
h2	2355	21	20	0.077s
h3	186	21	20	0.010s
h4	537	21	20	0.047s

6.

The time complexity of A* depends on the heuristic. In the worst case of an unbounded search space, the number of nodes expanded is exponential depending on the branching factor b which is equal to the average number of successor states. The heuristic function has a major effect on the practical performance of A* search, since a good heuristic allows A* to prune away many of unexplored states that an uninformed search would expand. Good heuristics have low effective branching factor.

As we see in q1, the number of states explored for a randomly generated grid varies for different heuristics. Heuristic h1 which is nothing but a normal bfs expands the most number of states while heuristic h2 performs better than h1 and h4 performs better than h2 and h3 is the most efficient of all heuristics.

h1: No of states explored = 164755

h2: No of states explored = 32598

h3: No of states explored = 1153

h4: No of states explored = 5800

In the A* algorithm, at each iteration, a node is chosen which minimizes the function:

$f(n)=g(n)+h(n)$

where $g(n)$ is the optimal cost from goal state to the current state n and $h(n)$ is the heuristic function that estimates the cost of the cheapest path from current state n to the goal state.

A heuristic function is said to be admissible if it does not overestimate the cost to reach the goal node. It estimates a cost to reach a goal that is smaller or equal to the cheapest path from n, which is denoted by $h^*(n)$. Therefore, an admissible heuristic h satisfies $h(n) \leq h^*(n), \forall n$. A* is guaranteed to find the optimal path if it uses an admissible heuristic.

Not all heuristics have the same information, hence all the admissible heuristics are not equally efficient. A more informed heuristic h3 is more efficient than a less informed heuristic h1 which contains no information but is still admissible.

Consider two versions of A*, each with a different admissible heuristic function

$f1(n)=g1(n)+h1(n)$ and,

$f2(n)=g1(n)+h2(n)$

where $h1(n) \leq h^*(n), \forall n$ and $h2(n) \leq h^*(n), \forall n$. Then A* with the evaluation function f1 is more informed than A* with f2 if, for all non-goal nodes n, $h1(n)>h2(n)$. Therefore more efficient.