Assignment-2 (A* Search)

Gaurav Kumar - 2211AI28

Ayush Kumar Singh - 2211AI27

Anurag Sharma - 2211AI29

Question:

In a general search algorithm each state (n) maintains a function f(n) = g(n) + h(n) where g(n) is the least cost from source state to state n found so far and h(n) is the estimated cost of the optimal path from state n to the goal state.

Implement a search algorithm for solving the 8-puzzle problem withfollowing assumptions.

A. g(n) = least cost from source state to current state so far.

B. Heuristics

a. h1(n) = 0. b. h2(n) = number of tiles displaced from their destined position. c. h3(n) = sum of Manhattan distance of each tiles from the goal position. d. h4(n) = Devise a heuristics such that $h(n) > h*(n)$.

1. Observe and verify that better heuristics expands lesser states.
2. Observe and verify that all the states expanded by better heuristics should also be expanded by inferior heuristics.
3. Observe and verify monotone restriction on the heuristics.
4. Observe un-reachability and provide a proof.
5. Observe and verify whether monotone restriction is followed for the following two Heuristics: 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 tiles from the goal position.
6. Observe and verify that if the cost of the empty tile is added (considering empty tile as another tile) then monotonicity will be violated.

Solution

A Search*

It is a searching algorithm that is used to find the shortest path between an initial and a final point.

It is a handy algorithm that is often used for map traversal to find the shortest path to be taken. A* was initially designed as a graph traversal problem, to help build a robot that can find its own course. It still remains a widely popular algorithm for graph traversal.

It searches for shorter paths first, thus making it an optimal and complete algorithm. An optimal algorithm will find the least cost outcome for a problem, while a complete algorithm finds all the possible outcomes of a problem.

Another aspect that makes A* so powerful is the use of weighted graphs in its implementation. A weighted graph uses numbers to represent the cost of taking each path or course of action. This means that the algorithms can take the path with the least cost, and find the best route in terms of distance and time.

Heuristic search

This technique is designed to solve problems quickly but does not guarantee an optimal solution but gives a good solution.

Manhattan distance

It is the summation of the absolute difference between of goal's x and y coordinates and current x and y coordinates.

Euclidean distance

It is the shortest distance from any node to the goal node.

*Algorithm for A search**

1. Initialize the open list
2. Initialize the closed list put the starting node on the open list (you can leave it at zero)
3. while the open list is not empty a) find the node with the least f on the open list, call it "q" b) pop q off the open list c) generate q's 8 successors and set their parents to q d) for each successor i) if successor is the goal, stop search ii) else, compute both g and h for successor successor.g = q.g + distance between successor and q successor.h = distance from goal to successor (This can be done using many ways, we will discuss three heuristics-Manhattan, Diagonal and Euclidean Heuristics) successor.f = successor.g + successor iii) if a node with the same position as successor is in the OPEN list which has a lower f than successor, skip this successor iv) if a node with the same position as successor is in the CLOSED list which h as a lower than successor, skip this successor otherwise, add the node to the open list end (for loop) e) push q on the closed list end (while loop)

Q.1) Observe and verify that better heuristics expands lesser states.

Ans : start state:

T1 T5 T6
T4 B T2
T7 T8 T3

goal state:

T1 T2 T3
T4 T5 T6
T7 T8 B

	Explored States	**Optimal Cost**	**Time taken**
Zero Heuristics	1617	12	0.1326465606689453
Tiles Displaced Heuristic	139	12	0.04886937141418457
Manhattan Heuristic	64	12	0.03989219665527344
Over Estimated Heuristic	115559	206	21.789729118347168

Here h3 heuristics expands least number of nodes. Hence h3 heuristics is better.

Q.2) Observe and verify that all the states expanded by better heuristics should also, be expanded by inferior heuristics.

Ans: All states are present in inferior.

Better heuristics : 64

Inferior heuristics :115559

To verify that all the states expanded by better heuristics we can use the intersection of sets. States expanded by better heuristics that are expanded by inferior heuristics: 64

Q.3) Q.5) Observe and verify whether monotone restriction is followed for the following two Heuristics:

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

Ans : Heuristic h1 : for every node h(n) <= cost(n,m)+h(m) Heuristic h2 : for every node h(n) <= cost(n,m)+h(m) Heuristic h3 : for every node h(n) <= cost(n,m)+h(m) Heuristic h4 : for every node h(n) <= cost(n,m)+h(m)

Q.4) Observe un-reachability and provide proof.

Ans: Total no. of nodes that can be reached: 9!/2 =181440 (approx to 50%)

To check unreachability we need to calculate inversions of each no. of starting matrix. If the sum is odd: not solvable If the sum is even: Solvable The matrix given in the question is Un-Solvable.

Un-Solvable.

T6 T7 T3
T8 T4 T2
T1 B T5

Q.6) Observe and verify that if the cost of the empty tile is added (considering empty tile as another tile) then monotonicity will be violated.

Ans : Yes ,its found that monotonicity will be violated after considering empty tile as another tile.

	Explored States	Optimal Cost	Time taken
Zero Heuristics	1617	12	0.1326465606689453
Tiles Displaced Heuristic	139	12	0.04886937141418457
Manhattan Heuristic	64	12	0.03989219665527344
Over Estimated Heuristic	115559	206	21.789729118347168

****With Empty Tile added****

	Explored States	Optimal Cost	Time taken
Zero Heuristics	1617	12	0.1565704345703125
Tiles Displaced Heuristic	149	12	0.04089021682739258
Manhattan Heuristic	67	12	0.0359034538269043
Over Estimated Heuristic	40261	150	8.683774709701538