CS250 - ARTIFICIAL INTELLIGENCE LAB

ASSIGNMENT-6 : A* Search

Date: February 14, 2024

Total Credit: 10

- Markings will be based on the correctness and soundness of the outputs.
- Marks will be deducted in case of plagiarism.
- Proper indentation and appropriate comments are mandatory.
- *All code needs to be submitted in '.py' format.* Even if you code it in '.IPYNB'format, download it in '.py' format and then submit
- You should zip all the required files and name the zip file as:
  - <roll_no>_assignment_<#>.zip, eg. 1501cs11_assignment_01.zip.

Problem Statement:

1. 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 with following 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 $h(n) > h*(n).$

Initial State:

1	2	3
4	5	6
7	8	B

Final State:

1	5	6
4	B	2
7	8	3

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.

Instructions:

1. You should make use of two lists for the implementation. One (close list) for maintaining the already explored states and another (open list) for maintaining the states which are found but yet to be explored.
2. Input is given in a file in the following format. Read the input and store the information in a matrix. Configuration of the start state and the goal state can be anything. For example given below T1, T2, ...,T8 are tile numbers and B is blank space.
3. Output should have the following information:

   a. On success:

i. Success Message
ii. Start State / Goal State

iii. Total number of states explored
iv. Total number of states to optimal path

v. Optimal Path
vi. Optimal Path Cost

vii. Time taken for execution

b. On failure:

i. Failure Message
ii. Start State / Goal State

iii. Total number of states explored before termination

4. Please make a table that should list the following for all the heuristics:

   - Total number of states explored.
   - Total number of states on optimal path
   - Optimal path.
   - Optimal Cost of the path.
   - Total time taken for execution

5. Please try to make your code as generic as possible (Preferably in Python).
6. Compare and contrast between the results of all four heuristics and state the reasons in a document file '*Why one heuristic is better than the other one?*'. While explaining, please comment on the optimality, time complexity etc.

For any queries regarding this assignment, contact:

Utsav Kumar Nareti (utsavkumarnareti@gmail.com)
Kumari Priya(kumaripriya.manit@gmail.com)
Akash Zingade(akashzingade@gmail.com)