CS561/571 - Executive Assignment

ASSIGNMENT-2: A* Search

Date: March 18, 2024

Deadline: April 1 , 2024

Total Credit: 10 (Implementation: 5; Documentation & Explanation: 5)

- 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.
- Upload your assignment (the zip file) to Moodle.
- **Note: Do not send your zip files to us by email**

Problem Statement:

- The assignment targets to implement A* search for the 8-puzzle problem

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 the 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 the following assumptions for the given start and end state. If unreachable, start with a random state and retry until the Target State (given above) is reached.

1. g(n) least cost from the source state to the current state.
2. Heuristics
    a. h1(n) = 0.
    b. h2(n) = number of tiles displaced from their destined position.
    c. h3(n) = sum of the Manhattan distance of each tile from the goal position.
    d. h4(n) = Devise a heuristics such that h(n) > h∗(n)

Tasks:

1. Observe and verify that better heuristics expand lesser states.
2. Observe and verify that all the states expanded by better heuristics should also be expanded by inferior heuristics.
3. Observe un-reachability and provide proof.
4. Observe and verify whether the monotone restriction is followed for the following two Heuristics:
    a. Monotone restriction: h(n) <= cost(n,m) + h(m)
    b. Heuristic:
        i. h2(n) = number of tiles displaced from their destined position.
        ii. h3(n) = sum of the Manhattan distance of each tile from the goal position.
5. Observe and verify that if the cost of the empty tile is added (considering the empty tile as another tile), then monotonicity will be violated.

Initial State (Sample)				Target State (Fixed)		
3	2	1		1	2	3
4	5	6		4	5	6
8	7	B		7	8	B

Instructions:

1. You should make use of two lists for the implementation. One (closed list) is for maintaining the already explored states, and another (open list) is for maintaining the states that are found but yet to be explored.

2. Input is given in a file, or the input is added to your code 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, T1, T2, …, and T8 below are tile numbers, and B is blank space.

3. Output file should have the following information:
    a. **On success:**
        i. Success Message

ii. Start State / Goal State
iii. The total number of states explored
iv. Total number of states to the 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

4. Total number of states explored before termination
5. Please make a table that should list the following for all the heuristics:
   a. The total number of states explored.
   b. Total number of states on the optimal path.
   c. Optimal path.
   d. Optimal Cost of the Path.
   e. Total time taken for execution
6. Please try to make your code as generic as possible
7. Compare and contrast the results of all four heuristics, h1(n), h2(n), h3(n), and h4(n), 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, etc.

For any queries regarding this assignment, contact:

Ratnesh Kr. Joshi (ratneshkr.joshi@gmail.com)