Lab Topic 03 - Solving 8-puzzle using A* Search

ZO Arnejo

A* Search

This algorithm is part of the family of search algorithms called **Informed Search**. The main characteristic of an informed search is its use of what is called **Heuristic value**. Using a heuristic function, a heuristic value is produced which gives insight to the algorithm in choosing the option that would likely lead to a solution as early as possible.

A*search has been used widely in the field of pathfinding and graph traversal. Its main approach involves the use of a heuristic plus an open list and a closed list. Open list contains all nodes that are candidates for examining (nodes identified as the frontier). Whereas, the closed list contains all nodes that have already been examined (nodes identified as explored). For 8-puzzle, the open list initially contains the starting position of the empty tile while the closed list starts with an empty content.

A* combines two values to compute for the estimated cost of the path through a node n to the goal node. The formula is given by:

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

The $g(n)$ is the exact cost of the path from the starting node to the node n. In 8-puzzle, it is the *total number of tile moves* from the start of the game to its current configuration.
The $h(n)$ is the **heuristic** estimated cost from node n to the goal node. The heuristic function is problem-specific. For 8-puzzle, we use a distance measure called **manhattan distance** given by:

$$distance = |x_1 - x_2| + |y_1 - y_2|$$

Manhattan distance gives insight in the distance of the individual tile to its supposed location. The final $h(n)$ is computed from summing all the manhattan distance of each tile, not including the empty tile.

An example computation of $h(n)$ is shown below:

2	1	3
5	4	
6	7	8

1	2	3
4	5	6
7	8	

2 1 3 5 4 6 7 8
1 2 3 4 5 6 7 8
---
1 1 0 1 1 3 1 1
= 9

The pseudocode is given by:

```
function AStar {
    openList = {initialState}    #frontier
    closedList = {}              #exploredList
    while(openList is not empty){
        bestNode = openList.removeMinF();
        closedList.add(bestNode);
        if(GoalTest(bestNode)) return bestNode
        for a in Actions(bestNode){
            x = Result(bestNode,a);
            if( (x is not in (openList or closedList))
                or (x is in openList and x.g < duplicate.g ) ):
                x.setParent(bestNode);
                openList.add(x);
        }
    }
}
```

Remember that the A* algorithm removes the node with the minimum value of *f,* which is computed as: $f = g + h$.

Exercise

Using your 8-puzzle game solved via BFS/DFS, add in your option A* Search. Also, provide a mechanism for selecting a new puzzle file in your game interface.

You are also required to submit a PDF file, created using LaTeX, that answers the following questions:

A. Compare the performance of breadth-first search (BFS), depth-first search (DFS), and A* for solving the 8-puzzle Game. You may include graphs that show the algorithm's difference in time spent, space occupied, etc.
   - Among the three, which is the most appropriate approach for solving the 8-puzzle game? Which is the least appropriate approach? Justify your answer.

B. Differentiate informed from uninformed search strategies. Compare their behavior and performance.

Scoring

Criteria	**Score**
Correct working interface for reading a new input file	2
Correct working interactive solutions interface	6
Write output file correctly	2
Documentation and Answers in LaTeX	5
Total	15

References:

Stuart Russell and Peter Norvig. 2009. *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.

Victor Adamchick, 2011. Homework Assignment 7 - Slide Puzzle. https://www.andrew.cmu.edu/course/15-121/labs/HW-7%20Slide%20Puzzle/lab.html Retrieved, 5 February 2020.