CS404 - ASSIGNMENT 1

Berkay Barış Turan - 28132

April 2023

Answers

1. In order to formulate a problem as a search problem, it is necessary to define the following components:

   - States: Colour-Maze puzzle instances have $n \times n$ size. In this assignment n values will be 10 so that we have a board of size 10×10.
     - Grid cells can have three different marks that are:
       - 0, denoting the cells that are empty and that need to be coloured,
       - X, denotes the cells occupied by the walls,
       - S, denotes the cell occupied by the agent, and
       - C, denotes the cell that is coloured, i.e., visited by the agent.
     - Therefore, our state space consists of (N(), meaning 'number of'):
       - $C(n^2, N(0)) \times C(n^2, N(X)) \times C(n^2, N(C)) \times C(n^2, N(S))$
   - Actions: The agent can move in four cardinal directions: up, down, right, or left.
   - Successor State Function: Given the current state, this function finds the possible actions that can be performed. Afterward, creates its successors corresponding for each action (finding their total costs, states, and action sequence that is performed to have the state of the successor from the initial state).
   - Initial State:
   - Goal Test: The agent's goal is to colour all the cells of the maze by moving over the cells while minimizing the total distance travelled by it. The function I defined in the Python implementation is checking the number of 0 in the matrix, if there exists no 0s, then the goal is achieved. However, for finding out the optimal case I am using priority queue so that the solution found is the one having the minimum cost.

- Step Cost Function: The cost of moving 1-unit of cell is equal to 1. In the implementation, I calculated the cost as the difference between the initial coordinates and the coordinates after the action performed.

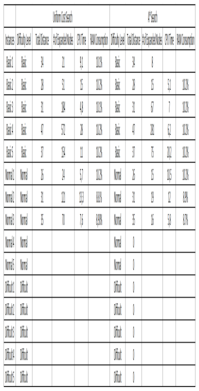
2. The Heuristic Function that I have implemented works as follows:

    - Heuristic function calculates the total number of 0s, i.e., the grid cells that have not been coloured by the agent, plus 1. The reason for adding 1 is that if all of the cells are coloured this function should return 1 because a heuristic function cannot return non-positive values.
    - I believe this heuristic function is admissible because it ensures that if there is an action that does not colour any grid (the agent is passing from the same place for no reason) it ensures that these actions are eliminated so that the total number of nodes expanded decreases drastically.
        - $f(n)$: f-value
        - $h(n)$: the value returned by heuristic function (always a positive number)
        - $g(n)$: cost of achieving the current state from the initial state.

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

3. For deciding on the difficulty levels, a simple way of assessment of directions is used. For instance, if there is only 2 ways the agent can move, the difficulty level of the maze has been decided as 'basic'. If there are 3 directions that the agent can go and should choose one of them, the difficulty level of the maze is decided as 'normal'. If there are any states that there are 4 directions that the agent can move, the difficulty level of that maze has been decided as 'difficult'.

4. To conclude up and compare the two search algorithms Uniform Cost Search and A* Search, it can be seen that the number of nodes expanded, the CPU time, and the memory usage of A* Search are clearly smaller than the values of Uniform Cost Search algorithm.



	Uniform Cost Search					A* Search				
Instances	Difficulty Level	Total Distance	# of Expanded Nodes	CPU Time	RAM Consumption	Difficulty Level	Total Distance	# of Expanded Nodes	CPU Time	RAM Consumption
Basic 1	Basic	34	21	9,1	10,1%	Basic	34	8		
Basic 2	Basic	28	51	15	10,2%	Basic	28	15	5,1	10,2%
Basic 3	Basic	31	184	4,8	10,1%	Basic	31	67	7	10,2%
Basic 4	Basic	47	572	28	10,2%	Basic	47	281	6,2	10,2%
Basic 5	Basic	37	154	11	10,2%	Basic	37	75	20,2	10,2%
Normal 1	Normal	26	24	5,7	10,2%	Normal	26	15	10,5	10,2%
Normal 2	Normal	31	121	13,3	8,8%	Normal	31	19	12	8,9%
Normal 3	Normal	35	70	7,6	8,90%	Normal	35	16	5,8	8,7%
Normal 4	Normal					Normal	0			
Normal 5	Normal					Normal	0			
Difficult 1	Difficult					Difficult	0			
Difficult 2	Difficult					Difficult	0			
Difficult 3	Difficult					Difficult	0			
Difficult 4	Difficult					Difficult	0			
Difficult 5	Difficult					Difficult	0			