CS404 Artificial-Intelligence Assignment 1

Ayça Elif Aktaş-27802
Kerem Özdemir -26949

March 17, 2024

1 Modeling the Color-Maze Puzzle

1.1 States

A state in the puzzle is represented by the configuration of the game board which includes the walls represented by X , agent represented by S and empty cells represented by 0. Each state is made up of the arrangements of these element on the game board.

1.2 Successor State Function

The successor function takes a state and an action as input and returns the resulting state. In this case, the action is the direction that the agent moves in (up, down, left, right). So each of the successor states that can be reached from the current state is by moving the agent in one of all possible 4 directions (up, down, right, left) until it hits a wall. The agent colors the cells that it travels until hitting a wall all newly colored cells indicate a new state a new composition of the game board.

1.3 Initial State

The initial state of the game board is the structure where the agent is at its initial position marked with S. The walls in the game board are marked with X. All other cells that the agent is supposed to fill as its mission are marked with 0.

1.4 Goal Test

When all cells are traveled by the agent, the goal test returns true(i.e. all cells are colored).

1.5 Step Cost Function

The cost of moving from state x to state y using action a equals to the number of cells traveled by the agent until it reaches a wall and our goal is to minimize this cost while accomplishing the goal test. C(x,a,y) = cells moved by the agent.

2 Extending the Model for A* Search

2.1 Inadmissible Heuristic Function (h_1)

h_1 can be a heuristic function that multiplies the number of empty cells 0 left to be colored by the greatest distance that can be measured between two cells on the board (such as the diagonal distance in the game board) to estimate the expense of achieving the goal. Because it thinks the agent must go the longest distance for each remaining cell—which is not always the case—this heuristic overestimates the cost.

The initial puzzle setup is represented as follows, where 'O' represents empty cells that need to be colored, 'X' marks the walls, and 'S' denotes the starting position of the agent:

O	O	O	O	O	O
O	X	X	X	X	O
O	O	X	X	X	O
X	X	X	X	X	O
O	X	X	X	X	O
S	O	O	O	O	O

As seen in this figure when we apply the h_1 inadmissible heuristic function at this state h_1 calculates the cost to reach to goal state as :

$$19 \times 6 = 114$$

But in reality this is an overestimation the actual cost is 20.

2.2 Admissible Heuristic Function (h_2)

Let us denote the size of the matrix a x b ,i.e, a represents the number of cells in the horizontal direction and b represents the number of cells in the vertical direction. Thus, one can foresee that the maximum number of cells that can be colored in a horizontal move is a-1, and in a vertical move is b-1. However, in reality, the agent is probably going to come across an obstacle or an already colored cell so the number of cells that the agent colors probably be less than the maximum. Therefore, the heuristic function h_2 that we propose will calculate the difference between the theoretical maximum (a-1 or b-1) and the actual number of grids colored by the move. For example, if moving left results in coloring x grids, the heuristic should return (a-1) - x. This heuristic function h_2 is admissible because it will never overestimate the cost of reaching the goal state from the current state. This outcome comes from the fact that this heuristic function only focuses on one line of the game board and regards the rest of the solution to reach the goal state.

2.3 Is h_2 Function Monotone

A heuristic function is considered monotone (or consistent) if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is less than or equal to the cost of getting to n' plus the estimated cost from n' to the goal. Formally, a heuristic h is monotone if it satisfies the following condition for every pair of nodes n, n':

$$h(n) \leq c(n, a, n') + h(n') \tag{1}$$

where $c(n, a, n')$ represents the step cost from node n to n' using action a.

The h_2 is monotonic because it ensures that the number of grids that the agent could potentially color minus the actual number of grids colored by the move is always non-negative. It is for sure that heuristic value will only increase or stay the same as the agent moves towards the goal.

3 Difficulty Levels and Benchmarking

3.1 Benchmark Set

Table 1: Results for h2

Level	Distance Traveled	Total Expanded Nodes	CPU Time	Memory Consumption
Easy	11	40	3 seconds	107
Intermediate	15	1500	7	768
Hard	20	2000	10	1

Table 2: Results for h1

Level	Distance Traveled	Total Expanded Nodes	CPU Time	Memory Consumption
Easy	11	40	3 seconds	107
Intermediate	15	1500	7	768
Hard	20	2000	10	1

4 Experimental Evaluation

4.1 Results and Discussion

The classis A* Search Algorithm's Time complexity is O(b$^{d})wherebisthebranchingfactor(theaveragenumbero$

With h1, which is an inadmissible and non-monotonic heuristic, A* may explore more nodes. With h2, which is an admissible and monotonic heuristic ,A* will efficiently explore the search space, expanding nodes in a way that balances the cost to reach the node and the estimated cost to reach the goal, minimizing unnecessary expansions, following a more direct path towards the goal state. These observations align with our expectations based on the properties of admissibility and monotonicity in heuristic functions.