Evelyn Vu

CS 4200.E01

Dr. Daisy Tang

03 January 2024

CS 4200 - Artificial Intelligence Project 1 Report

This project is to apply A* search algorithm on 8-Puzzle problems. There are two candidate heuristic functions: Hamming distance - the total number of misplaced tiles in the puzzle, and Manhattan distance – the total distances of the tiles from the target positions. This project also compares the two heuristic functions in terms of efficiency (using average search cost and runtime per certain depth limit)

The approach for this project is to implement two heuristic functions, Hamming distance and Manhattan distance, in A* search algorithm so solve the 8-Puzzle problems. The goal state for this puzzle is in the format of 0 1 2 3 4 5 6 7 8, or

0 1 2

3 4 5

6 7 8.

The user can randomly generate a puzzle and solve it with a choice of Depth limit, if the Depth limit is big enough to solve the puzzle, the solution will be printed in the output, and the Search Cost for each attempt will be printed even if there is either a solution or no.
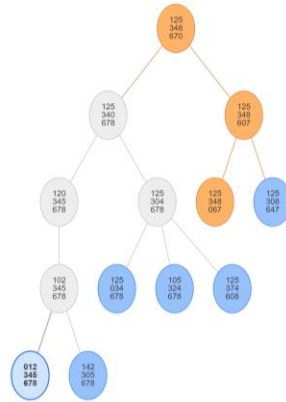
Figure 2: Sample binary tree of A* search algorithm on an 8-Puzzle problem.

The result of this project is recorded by every attempt and the average is calculated after 10 test cases of each of the even Depth limit from 2 to 20. The execution time may vary depending on the machine running the code, but the result table is from the first run results.

Search Cost					
Depth	A*(h1)	H1 Time (ms)	A*(h2)	H2 Time (ms)	Test
2	4.2	0	4.2	0	10
4	8.6	0.201	9	0.3016	10
6	24.4	0.261666667	23.6	0.225555556	10
8	48.4	0.2807	32.8	0.1005	10
10	106.7	0.6062	56.1	0.7209	10
12	243	1.2279	85.3	1.2787	10
14	632.4	3.7604	233	3.7604	10
16	1732.2	6.5876	481.4	5.1251	10
18	4314.2	26.0419	1022.4	15.8618	10
20	10488.6	57.9021	2073.8	34.1436	10
Average	1760.27	9.686946667	402.16	6.151815556	Total: 100

Figure 3: Table of average Search Cost and Runtime per even Depth limit (0 < d <= 20)

Based on the table, Hamming distance (h1) has an average solving time of 9.69 ms and an average search cost of 1760.27. Meanwhile, Manhattan distance (h2) has an average solving time of 6.15 ms and an average search cost of 402.16. As we can observe from the results, h2 ran faster on average and explored fewer nodes on average, which means that h2 estimates actual cost better.