Project 1: An Evaluation of Uninformed and Informed Search Algorithms on the k-puzzle Problem

Wang Zicong A0184475U, Yang Zi Yun A0184682U, Yuan Jiayi A0177893H, and Zhang Yihan A0177861R

National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077

1 Problem specification

1.1 State

Each state is represented as a 2d array of size k*k. The empty cell is represented by the number 0.

1.2 Actions

The valid moves in the k-puzzle problem: the movement of a numbered cell to a neighbouring cell that is empty. The actions are {Left, Right, Up, Down}, which is to move a cell left/right/up/down for 1 step to occupy the empty cell.

1.3 Transition Model

T(current state $S_{current}$, possible actions) = {next state S_{new}}

Swap the numbered cell with its neighbouring empty cell in the direction given by the input action. An example of state transition model is shown in Figure 1 below.

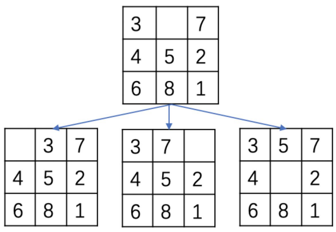

Fig.1. Transition model of state

1.4 Goal Test

All the cells are in ascending order, and the empty cell is the last cell.

1.5 Cost Function

1 for each step.

2 Technical analysis

2.1 Uninformed Search

Complete	Yes, since b (branching factor, number of valid moves) is finite.
Optimal	Yes, since the cost of each step is 1.
Time complexity	$O(b) + O(b^2) + O(b^2) + \ldots + O(b^d) = O(b^d)$
Space complexity	Max size of frontier $O(b^d)$, d is the depth of shallowest goal, which is the optimal goal.

2.2 **Informed Search**: A* Search using graph-search

Complete	Yes, since b (branching factor, number of valid moves) is finite.
Optimal	Yes, since each of the heuristics are consistent.
Time complexity	$O(b^{h^*(s0) - h(s0)})$
Space complexity	Max size of frontier $O(b^m)$, m is the maximum depth of search tree

2.3 Proof of consistency:

Heuristic 1: Hamming distance/ Number of misplaced tiles

- Admissible and consistent (proof omitted)

Heuristic 2: Manhattan distance

- Admissible and consistent (proof omitted)

Heuristic 3: No. of tiles not in goal row + No. of tiles not in goal col

- We derive this heuristic by the relaxed k-puzzle problem: the cell can move to any square that is at the same row or the same column in one step.
- Explanation: Consider the state *s* in Figure 2 below, the number “1” takes 2 steps to reach its goal position. The number “3” takes 1 steps to reach its goal position. Hence, h(s) = 2 + 2 + 1 + 0 + 0 + 2 + 2 + 0 +2 = 11

3		7
4	5	2
6	8	1

Fig.2. State s

- Proof of consistency:
  From state n to its neighboring state n', there are 3 possible values for h(n'). h(n') = h(n) - 1 if a tile moves into its goal row or column compared to it in state n. h(n') = h(n) if a tile is still not in its goal row or column after moving from n. h(n') = h(n) + 1 if a tile moves out from its goal row or column. Therefore, h(n') >= h(n) - 1, that is h(n) <= h(n') + 1 = h(n') + c(n, n').
  Hence, h3 is consistent.

3 Experimental setup

3.1 Goal of the experiments

- To compare the time and space complexity of the chosen algorithm and heuristics. To test the performance of chosen heuristics under K-puzzles with different depth of goal.

3.2 Experimental Setup

- Input: 3-puzzles of different depth of goal
- Output: number of visited nodes for each search algorithm
- To replicate this experiment, the command is *python CS3243_P1_30_5.py test_cases/* , the results can be found in the *experiment_result.txt* file.

3.3 Experiment result

Table 1. Number of visited nodes

Depth of goal	BFS	Heuristic 1	Heuristic 2	Heuristic 3
10	271	31	20	10
14	2395	188	85	103
20	34464	2769	401	1067
24	113404	18069	1177	5569
28	175303	65262	4835	22114
31	181422	123490	11664	55111

4 Results and Discussion

4.1 Results and comparison of performances of the chosen algorithms and heuristics

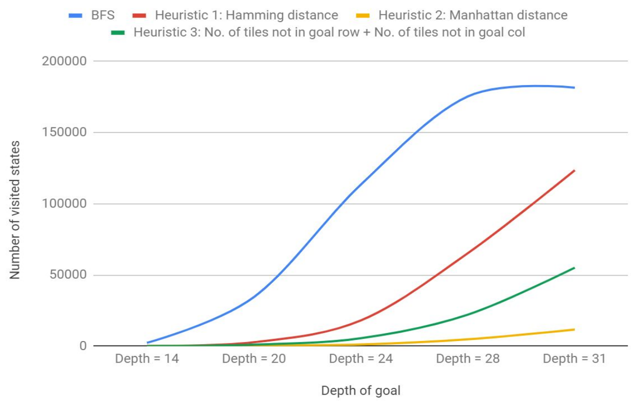

Fig.3. Number of visited states of different heuristics

From Table 1 and Figure 3 in experiment result, we can conclude that the number of visited states of the heuristics are: Manhattan Distance < Heuristic 3 < Hamming Distance < BFS for the same depth of goal. Hence, vise versa, Manhattan Distance Heuristic has the fastest and more optimal speed compared to others while solving the same puzzle.

Hence, Manhattan Distance dominates Heuristic 3, and Heuristic 3 dominates Hamming Distance.

4.2 Performance of the chosen algorithms and heuristics under puzzles with different depth of goal

From the experiments, we noticed a drop of time and space taken for the same algorithm under higher depth of goal. We believe this is because the depth of goal does not directly determine the difficulty of solving the puzzle. Instead, it is a general trend that as depth of goal increases, difficulty or, time and space complexity increases correspondingly. This is because the search process would have to go layers deeper in order to find the goal state, while the position of the goal state inside the layer is undetermined.