DA221 Assignment 1 Report : 8 Puzzle

By Himanshu Singhal
220150004

1 Introduction

This report presents the solution to the lab assignment 1 of the course DA221 that asks for a program to solve the 8-puzzle problem using the A* algorithm with the help of various heuristic functions.

1.1 Formulating the Problem

The 8-puzzle consists of an area divided into a grid, 3 by 3 for the 8-puzzle. On each grid square is a tile, except for one square which remains empty. Thus, there are eight tiles in the 8-puzzle. A tile that is next to the empty grid square can be moved into the empty space, leaving its previous position empty in turn. Tiles are numbered, 1 thru 8 for the 8-puzzle, so that each tile can be uniquely identified.

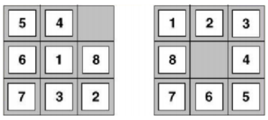

Figure 1: *An instance of the 8-puzzle problem*

2 The A* algorithm

I have used the A* algorithm here to find the solution to this problem. It evaluates nodes by combining $g(n)$ (the cost to reach the node) and $h(n)$ the cost to get from the node to the goal:

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

Now the next section will discuss the 3 heuristic functions

3 Heuristic Functions

In my implementation, I have used three heuristic functions: *Hamming Priority*, *Manhattan Distance*, and *Manhattan with Linear Conflict* out of the 3 heuristics, Manhattan with Linear Conflict performed the best. All of which are discussed in the upcoming subsections.

3.1 **Hamming Heuristic**

The Hamming Heuristic is defined by the number of misplaced tiles in the 8-puzzle. By its nature, we can say that this function is **admissible** because it very naively gives the minimum number of moves required to move an out-of-place tile to its correct position without accounting for the complexity of moving other tiles around to get the single out-of-place tile into place. Because of this, his function either underestimates or gets the accurate number of moves needed but never *overestimates*.

Hence, this function gives a lower bound for the number of moves required to get to the goal state.

3.2 **Manhattan Distance**

The Manhattan distance measures the distance between 2 points in a grid-based layout. The name ”Manhattan” is inspired by the streets of Manhattan, which have a grid-like layout. It is calculated as the sum of the absolute difference of their Cartesian coordinates. In other words, it computes the horizontal and vertical moves needed to reach the target position from the current position.

The Manhattan heuristic is admissible since it never overestimates the cost to get from the current state to the goal state. It is impossible to solve to the tile's position in fewer moves than the Manhattan distance, which ensures that the A* algorithm using this heuristic will always find the shortest possible solution.

3.3 **Manhattan with Linear Conflict**

Manhattan with Linear Conflict is an addition to the Manhattan distance which improves its accuracy. This approach not only considers the individual Manhattan distance of the tiles but also recognizes specific configurations of the tiles that can increase the number of moves required to reach the goal state.

Linear conflict is an additional factor that comes into play when two tiles are in their correct row/column but in the wrong order, and one must move out of the way for the other to reach its goal position. For each pair of such conflicting tiles, at least two extra moves are needed - one to move a tile out of the way and another to move it back into place once the blocked tile reaches its correct position.

By considering these linear conflicts, this heuristic more accurately reflects the complexities of the puzzle, resulting in a more precise estimation of the real distance to the goal.

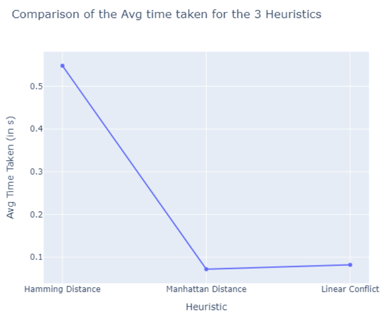

Figure 2: *Avg time taken for the 3 heuristics*

4 Evaluation of the Heuristics

4.1 Hamming Heuristic

- **Accuracy**: Low, as it only counts the number of tiles in the wrong position, without considering their distance from the correct positions.
- **Computational Complexity**: Low. It is Easy and fast to compute as it only requires checking whether each tile is in its correct position.
- **Search Efficiency** Low. Since it provides a very basic estimate, it can lead to a larger search space and a longer time to find the solution.

4.2 Manhattan Distance

- **Accuracy**: Medium. Calculates the sum of the distances of each tile from its goal position, assuming direct paths.
- **Computational Complexity**: Moderate. Requires calculating the Manhattan distance for each tile, which is more computationally intensive than the Hamming heuristic.
- **Search Efficiency**: Moderate. Provides a more accurate estimate than the Hamming heuristic, leading to a more efficient search in most cases.

4.3 Manhattan with Linear Conflict

- **Accuracy**: High. It includes the Manhattan distance and additionally accounts for linear conflicts, offering a closer estimate of the actual number of moves required.
- **Computational Complexity**: Higher. It involves calculating both the Manhattan distance and detecting linear conflicts, which adds to the computational overhead.
- **Search Efficiency**: High. By providing a more accurate heuristic, it significantly reduces the search space, often leading to faster solutions.

5 Experimental Results

Given below are 10 randomly generated instances of the 8-puzzle:

[[7 1 5]	[[0 1 5]	[[8 2 0]	[[1 7 6]	[[0 5 2]
[0 8 2]	[3 4 7]	[6 7 5]	[2 8 0]	[6 3 7]
[4 3 6]]	[8 6 2]]	[3 1 4]]	[3 4 5]]	[4 1 8]]
[[8 5 1]	[[2 8 4]	[[3 1 4]	[[7 0 1]	[[5 1 3]
[7 0 2]	[1 0 3]	[5 7 6]	[3 2 4]	[7 6 4]
[4 3 6]]	[6 5 7]]	[8 0 2]]	[5 8 6]]	[8 0 2]]

Figure 3: *10 Randomly Generated Instances*

And here is the goal state:

[[0 1 2]
[3 4 5]
[6 7 8]]

Figure 4: *The Goal state*

All the statistics presented here are based on these 10 instances.

Heuristic Used	Avg time Taken (in s)	Avg Length of Solution	Avg Number of Nodes Taken Out
Hamming	0.548	23	11395
Manhattan	0.07191	23	1451
Linear Conflict	0.08198	23	810

Table 1: *Experimental Results on the 10 random instances*

Figure 5: *Comparision between the Avg Number of nodes taken out from the Frontier*

6 References

- Artificial Intelligence: A Modern Approach 3rd Edition by Stuart Russell Peter Norvig
- Princeton page on 8-Puzzle
- Wikipedia page on A* algorithm