

# CSCI 4511W Writing 5

Sid Lin (linx1052) and Noah Park (park1623)

April 3, 2020

One of the most common introductory topics to AI are search algorithms. Search algorithms are one of the most basic and familiar topics for computer scientists beginning to expand their learning into the realm of AI. These algorithms can be split into two categories, uninformed search and informed search. Many search algorithms have practical applications with familiar problems, such as the 8 queens, traveling salesperson problem, map coloring,  $N \times N$  slide puzzle, and much more. For our project, we will be focusing on the 8-puzzle with informed search.

The 8-puzzle scenario is none other than the familiar slide puzzle; each state consists of a  $N \times N$  grid with all numbered tiles, aside from a single missing one. The goal of the puzzle is to move each tile into a certain order with the fewest moves and time. It may appear as if the moves are made by moving the numbered tiles into the empty space, but a simpler explanation is that the empty tile moves around the board. A board that is  $3 \times 3$  will have  $9!$  possible permutations, but only half of them are solvable. The sheer amount of possible permutations make the 8-puzzle is an ideal candidate for an informed search analysis [2].

Each configuration of the 181,440 possible instances of the 8-puzzle was tested with an optimal iterative deepening algorithm [3]. Richard Korf's IDA\* algorithm was used in conjunction with the Manhattan distance heuristic to solve each configuration. Each solution had more than one optimal solution, but the average number of steps to solve the puzzle was consistent. The maximum number of steps to solve the 8-puzzle was 31, and the average was about 22 [3].

One of the most popular informed search algorithms is A\*. It is an extension of Dijkstra's algorithm. The A\* algorithm uses a cost function that takes both path cost and heuristic cost into account. Normally, A\* search spaces are implemented with nodes, which can have children and a predecessor node. Most importantly, A\* uses a priority queue ordered with the lowest function cost to determine each node's order of expansion. The standard function is denoted  $f(x) = g(x) + h(x)$ , where  $g$  and  $h$  are the path cost and heuristic cost respectively [1]. The optimality of this algorithm depends on the admissibility of the heuristic function as well.

Our implementation of the 8-puzzle was done in Swift and published on the iOS App Store. To measure the effectiveness of the A\* search algorithm, we

had a board randomization function with 4 levels of difficulty, correlated with the number of random moves made.

## References

- [1] Masoud Nosrati, Ronak Karimi, and Hojat Allah Hasanvand. Investigation of the\*(star) search algorithms: Characteristics, methods and approaches. *World Applied Programming*, 2(4):251–256, 2012.
- [2] Rok Piltaver, Mitja Luštrek, and Matjaž Gams. The pathology of heuristic search in the 8-puzzle. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):65–94, 2012.
- [3] Alexander Reinefeld. Complete solution of the eight-puzzle and the benefit of node ordering in ida\*. In *International Joint Conference on Artificial Intelligence*, pages 248–253, 1993.