# HW3 8-Puzzle Analysis

Sam Badger & Sean Farris (Group 9)

For our analysis, we created a program that randomly generated puzzles and solved them using the following 4 heuristics:

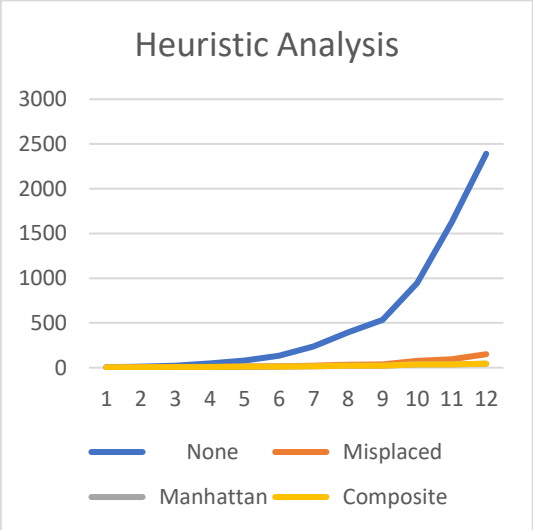
1. No heuristic (Always estimates 0)
2. Misplaced Tiles
3. Manhattan Distance
4. Composition of (2) and (3) (always gives maximum between the two).

Pseudo code for the heuristic analysis program is shown below. As you can see, we increase the difficulty of the randomly generated puzzle to be solved. We generate 10 puzzles of each difficulty and solve them using each heuristic, then report the average number of states explored for each heuristic with each difficulty level.

```
for (DIFFICULTY in range [1,20]):
      for (10 iterations):
            RAND_PUZZLE = puzzle that can be solved with DIFFICULTY steps.
            for (HEURISTIC in all heuristics):
                  solve RAND_PUZZLE with A* algorithm using HEURISTIC.
                  record the number of states explored.
      report average number of states explored for each heuristic.
```

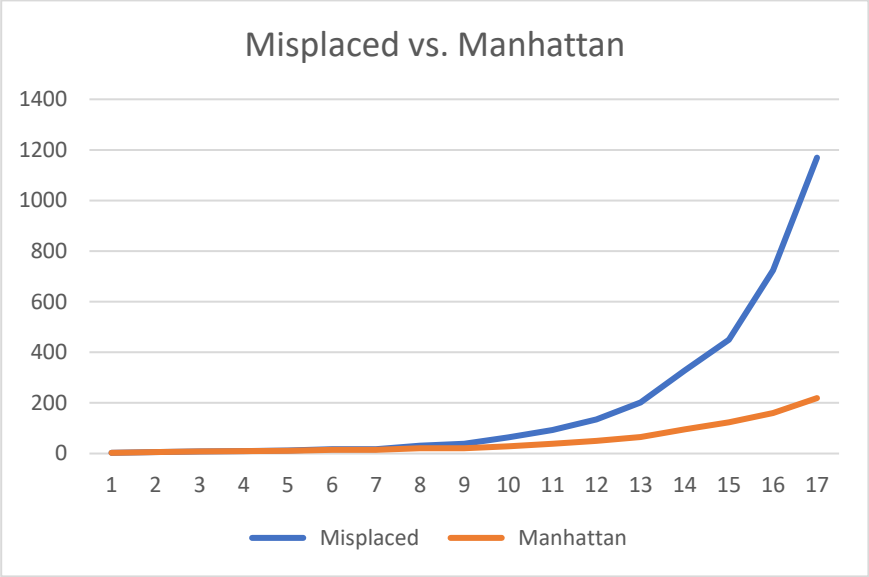
An example output is shown below:

| diff | None | Misplaced | Manhattan | Composite |
|---|---|---|---|---|
| 1, | 4.900, | 2.900, | 2.900, | 2.900, |
| 2, | 13.200, | 6.200, | 6.200, | 6.200, |
| 3, | 23.600, | 7.100, | 7.100, | 7.100, |
| 4, | 46.300, | 9.800, | 9.400, | 9.400, |
| 5, | 79.300, | 11.500, | 11.200, | 11.200, |
| 6, | 133.700, | 14.700, | 12.900, | 12.900, |
| 7, | 237.600, | 20.900, | 16.300, | 16.300, |
| 8, | 393.200, | 33.500, | 22.200, | 22.200, |
| 9, | 533.000, | 38.200, | 21.000, | 21.000, |
| 10, | 946.200, | 75.500, | 36.700, | 36.700, |
| 11, | 1624.500, | 94.000, | 36.100, | 36.100, |
| 12, | 2390.500, | 150.300, | 44.300, | 44.300, |




The chart shows that in general, any reasonable heuristic is better than no heuristic. Because using no heuristic takes forever to run and we aren't patient enough, we run again without the No heuristic. We also notice that the Composite heuristic always performs the same as the Manhattan Distance heuristic, so we remove that one as well. The results are shown below:

| diff | Misplaced | Manhattan |
|---|---|---|
| 1, | 2.800, | 2.800, |
| 2, | 5.800, | 5.800, |
| 3, | 7.900, | 7.900, |
| 4, | 10.000, | 9.800, |
| 5, | 12.600, | 11.300, |
| 6, | 16.600, | 14.500, |
| 7, | 17.400, | 14.900, |
| 8, | 31.100, | 20.600, |
| 9, | 38.500, | 20.700, |
| 10, | 63.500, | 28.300, |
| 11, | 92.800, | 38.100, |
| 12, | 135.000, | 49.700, |
| 13, | 202.300, | 65.300, |
| 14, | 328.400, | 95.400, |
| 15, | 449.300, | 122.800, |
| 16, | 724.700, | 159.700, |
| 17, | 1169.400, | 219.100, |




As you can see, the Manhattan distance heuristic performs much better with harder puzzles.

## Conclusion

Any heuristic that provides a reasonable underestimate of the distance remaining from the current state to the goal state is better than using no heuristic. The best heuristic is the one that gives the most accurate approximation. For any heuristic to guarantee us the optimal path, it must not provide an over-estimate.

The heuristic that provides the most accurate estimate is the one that provides the maximum underestimate – thus, if a variety of heuristics are available, the maximum value of all heuristics can be used.