# Manhattan Distance V.S. Misplaced Tiles Heuristics

## *Solving 8 Block Sliding Puzzle using A* Search*

Assignment 1
Jason S. Wuertz
119089
CSC 411
Qiang Ye
02/21/14

# Part I

Problem instances were initially created using true randomness. That is an array with the numbers 0 through 8, inclusive, was instantiated. This array was then shuffled using Java's Collections library. The shuffled array was then inserted into a two dimensional array representing the puzzle. This resulted in a truly random instance of the problem. However this also resulted in problems who's solution depth was usually above 24, meaning that a search could take hours to complete. In order to avoid this another method was devised to create problem instances.

The constructor for TilePuzzle, the problem instance, takes a boolean parameter indicating whether true randomness will be used to instantiate the object. If false, the object can be shuffled after creating using a circular shuffling technique. This technique can be called on an object via the shuffle function which takes a integer parameter. This parameter defines the approximate depth to which the puzzle will be shuffled. Circular shuffle works by moving the blank space around a circle by exchanging it with it's adjacent squares either clock wise, or counter-clock wise, depending on which corner it is currently in. This can be seen in Figure 1 in the appendix. The algorithm then alternates between the four corners *n* amount of times, where *n* is the integer defined by the parameter.

# Part II

The test results of the two problems were can be seen in Figure 2 in the appendix. It would be expected that the Manhattan Distance heuristic would outperform it's counterpart exponentially. My experiments indicate that the heuristic is slightly less efficient at solving small problems, which may indicate that there is an error when computing the heuristic value. However the Manhattan Distance heuristic does outperform the Misplaced Tiles heuristic once the problem reaches significant depth. Using the data found in the Russell and Norvig, Artificial Intelligence: A Modern Approach, it can also be seen that my Manhattan Distance heuristic generates significantly more nodes than it should. This proves that there is a flaw in my calculations or when counting the number of nodes generated. Even with this flaw it can be seen that Manhattan Distance dominates Displaced Tiles since for any node *n*, *h1(n)>h2(n)*, where *h1* is the Misplaced Tiles heuristic and *h2* is the Manhattan Distance heuristic.

Due to the way I generated the problems, the algorithm actually never generated problems of any other depths other than the ones displayed in the table. This is flawed since it doesn't allow for more detailed comparison of the heuristics. In addition there is no value for depth 26 for *h1* (Misplaced Tiles) since it took too long to solve the problem on my computer. This displays that *h1(26)* generates a significant number of nodes.

## Appendix

Figure 1

| | | |
|---|---|---|
| ↓ | ← | X |
| → | ↑ | X |
| X | X | X |

| | | |
|---|---|---|
| X | → | ↓ |
| X | ↑ | ← |
| X | X | X |

| | | |
|---|---|---|
| X | X | X |
| → | ↓ | X |
| ↑ | ← | X |

| | | |
|---|---|---|
| X | X | X |
| X | ↓ | ← |
| X | → | ↑ |

Figure 2

| Depth | Misplaced Tiles(h1) | Manhattan Distance(h2) |
|---:|---:|---:|
| 4 | 14 | 15 |
| 8 | 38 | 44 |
| 14 | 519 | 411 |
| 18 | 3434 | 1766 |
| 20 | 6864 | 3641 |
| 24 | 34755 | 15352 |
| 26 | - | 31210 |