Luis Miguel Flores

2015 – 01835

CS 180 MP1 ANALYSIS

CPU: Intel Core i7

RAM: 8 GB

1. Number of Configurations Generated

- Case 1:
  - ➢ Misplaced Tiles - 276
  - ➢ Linear Conflict - 166
  - ➢ Tiles out of Row and Column - 8
  - ➢ Nmax Swap - 12
  - ➢ Manhattan Distance - 8
- Case 2:
  - ➢ Misplaced Tiles - 181439
  - ➢ Linear Conflict - 179921
  - ➢ Tiles out of Row and Column - 47699
  - ➢ Nmax Swap - 90876
  - ➢ Manhattan Distance - 7131
- Case 4:
  - ➢ Misplaced Tiles - 277
  - ➢ Linear Conflict - 154
  - ➢ Tiles out of Row and Column - 6
  - ➢ Nmax Swap - 6
  - ➢ Manhattan Distance – 6

2. Runtime

- Case 1:
  - ➢ Misplaced Tiles – 0.135s
  - ➢ Linear Conflict – 0.167s
  - ➢ Tiles out of Row and Column – 0.143s
  - ➢ Nmax Swap – 0.169s
  - ➢ Manhattan Distance – 0.136s

- Case 2:
  - ➢ Misplaced Tiles – 4.408s
  - ➢ Linear Conflict – 5.610s
  - ➢ Tiles out of Row and Column – 1.702s
  - ➢ Nmax Swap – 3.422s
  - ➢ Manhattan Distance – 0.415s
- Case 4:
  - ➢ Misplaced Tiles – 0.174s
  - ➢ Linear Conflict – 0.154s
  - ➢ Tiles out of Row and Column – 0.152s
  - ➢ Nmax Swap – 0.134s
  - ➢ Manhattan Distance – 0.141s

3. Proof of admissibility of the heuristic functions

- Misplaced Tiles
  - ○ Let L be reachable state
  - ○ Assume h(L+1) <= g(L+1) where h(L+1) is the heuristic cost from L to goal state, and g(L+1) is the actual cost
  - ○ Prove true for L
  - ○ h(L) is at most equal to h(L+1) + 1
  - ○ g(L) = g(L+1)+1
  - ○ h(L+1) <= g(L+1)
  - ○ h(L+1) + 1 <= g(L+1) +1
  - ○ h(L) <= g(L)
- Linear Conflict
  - ○ At best, it takes 4 moves to solve a linear conflict. For each pair in the linear conflict, it is 2 + the manhattan distance of the tiles in linear conflict. This means that the actual cost is greater than or equal to the manhattan distance of the two tiles plus the heuristic of the linear conflict (2 for each pair). Linear conflict heuristic is less than the manhattan distance + the linear conflict heuristic. Therefore, the linear conflict heuristic is less than or equal to the actual cost

- Tiles out of row and column
    - A tile is included in the out of row and/or out of column counter regardless of how far from its goal state it is. Essentially, a tile out of row is given the same count as the manhattan distance of a tile next to its goal state horizontally, and a tile out of column is given the same count as the manhattan distance o a tile next to its goal state vertically. Therefore, the heuristic given by the number of tiles out of row and column is always less than or equal to the heurisic given by the manhattan distance. We know that the manhattan distance heuristic is always lesst than or equal to the actual cost because it is admissible. Therefore, the heuristic given by the number of tiles out of row and column is also less than or equal to the actual cost.
- Nmax Swap
    - The actual cost is calculated by the number of moves made before reaching goal state. The number of moves for the actual cost is restricted to those adjacent to the empty space which is at most 4 moves. Nmax swap moves are not restricted by this and can have as many as (n*n)-1 moves in attempting to reach the goal state. Nmax swap moves are therefore a best case scenario for the normal board states, wherein the Nmax swap moves are those that can be done in a normal board state. Therefore, the heuristic for Nmax swap is less than or equal to the actual cost.
- Manhattan Distance
    - The Manhattan Distance heuristic is calculated by getting the sum per tile of how many moves a tile would need to make in order to get to its goal state assuming there were no other tiles (but still following the rule of only vertical and horizontal movements). Also note that moving a tile does not change the positions of other tiles (essentially it swaps places with the empty space). This means that h is the number of moves necessary for each tile to reach their goal states without any interferences, which is the best case scenario for ordinary board states. Being the best case scenario, h is either less than or equal to g for any board state.

4. Best heuristic: Manhattan Distance
5. Without Heuristics
    - Case 1
        - Runtime – 0.145s (Better than Linear Conflict and Nmax Swap but negligible)
        - Configurations – 177 (Better than Misplaced Tiles)

- Case 2
    - Runtime – 4.517s (Better than Linear Conflict)
    - Configurations – 181438 (Same as Misplaced Tiles, worst of all)
- Case 4
    - Runtime – 0.162 (Better than Misplaced Tiles but negligible)
    - Configurations – 154 (Same as Linear Conflict, better than Misplaced Tiles)