Artificial intelligence Lab: CSE-402

Report on Offline 2: N puzzle with A-star algorithm

Prepared by

Md. Salman Estyak,

Student No: 1105094

Heuristics used for this assignment are-

1. Tiles out of row and column heuristic. (h1)
2. N-Max Swap or Gasching’s heuristic (h2)

**Admissibility:**

The heuristic for N-Max Swap, or Gasching’s heuristic, h2 is the number of steps it would take to solve the problem if it was possible to swap any tile with the "space".

In the original problem, we can only swap the empty tile with another tile that’s directly adjacent to it horizontally or vertically.

But, in Gasching’s heuristic, we can swap the empty tile with any other tile. Hence, this is an easier problem.

And, this underestimates the actual distance function. Hence, this is admissible.

Tiles out of row and tiles out of column is also an admissible heuristic, since every tile that is out of column orout of row must be moved at least once and every tile that is both outof column and out of row must be moved at least twice

From Table-1 we can observe that, the nodes explored with heuristic-2 is always greater (most of the time, it explored twice the number of nodes explored by heuristic-1) than heuristic-1. Hence heuristic-1 provides a closer lower-bound on the to the actual cost path to reach the goal, i.e., it is more accurate, hence it is more admissible.

**Comparative performance:** (For 8 puzzle, 0 stands for the blank tile)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Start State | Goal State | Nodes Explored, N\_h1 for h1 | Nodes Explored, N\_h2 for h2 | N\_h2/N\_h1 | Number of moves | Run time for h1 (s) | Run time for h2 (s) |
| 724631805 | 123456780 | 2495 | 9660 | 3.87 | 23 | 0.234 | 1.719 |
| 206134758 | 123456780 | 17 | 30 | 1.76 | 9 | 0.002 | 0.016 |
| 806547231 | 012345678 | 53559 | 83678 | 1.56 | 31 | 5.85 | 17.237 |
| 724506831 | 012345678 | 10330 | 27192 | 2.6 | 26 | 1.039 | 5.194 |
| 261078354 | 123456780 | 13836 | 20552 | 1.48 | 27 | 1.419 | 3.817 |
| 012345678 | 123456780 | 2178 | 4734 | 2.17 | 22 | 0.516 | 1.016 |
| 543018762 | 127803456 | 181440 | 181440 | 1 | N/A | 21.994 | 44.120 |
| 1 2 3 4  5 6 7 8  9 10 11 13  15 14 12 0 | 1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 0 | 14885 | 29857 | 2.00 | 24 | 2.973 | 9.88 |
| 4 1 2 3  5 6 7 11  8 9 0 10  12 13 14 15 | 0 1 2 3  4 5 6 7  8 9 10 11  12 13 14 15 | 7 | 7 | 1 | 6 | 0.332 | 0.084 |
| 2 3 4 0  1 5 7 8  9 6 10 12  13 14 11 15 | 1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 0 | 10 | 10 | 1 | 9 | 0.142 | 0.085 |

Table 1: Comparative performance analysis of the two heuristics.

**Discussion:**

From the above data, we can see that for many (start\_state, goal\_state) pairs, we get small number of nodes explored and for some other pair the algorithm explores a lot of nodes.

And, during experimentation, some pairs of start state and goal state was encountered for which the algorithm could not find any solution, later on it was discovered that for some particular goal states and start states, the algorithm might never be able to reach the goal state , these pairs are unsolvable pairs. For instance, (812043765, 123456780) is such an unsolvable pair. For one pair (543018762, 127803456) the algorithm terminated for both heuristic functions, both explored nodes but they did not come up with any solution, I think it’s because the whole search space was traversed so the algorithm ended but, as the goal state was unreachable, hence we didn’t any get solution.

For some inputs, the number of nodes explored by both of the heuristics were the same, but in those cases, the run time of the second heuristic was better.