CSC 831: Artificial Intelligence

Solving 8-Puzzle Game Using Steepest Ascent Hill Climbing Algorithm

EDAWARE PATRICIA EWOMA-ZINO
179074012

ABSTRACT

The 8-Puzzle game is a difficult puzzle to solve and the largest puzzle of its type that can be completely solved. It should be noted however that not all 8-puzzle board configuration can be solved, solvable and unsolvable board configuration can be detected through a method called inversion. However, by applying specific informed search methods in the field of Artificial Intelligence, the problem can be solved. In this work, the 8-puzzle game was solved using Steepest Ascent Hill Climbing Algorithm by applying the Manhattan distance heuristics, which checks for the difference between a tile position on a current board configuration and the goal board configuration.

INTRODUCTION

The Problem Definition

The 8-puzzle consists of an area divided into a 3 x 3 grid. On each grid square is a tile, expect for one square which remains empty. Therefore, there are eight tiles in the 8-puzzle game. A tile that is next to the empty grid square can be moved into the empty space, leaving its previous position empty in turn. Tiles are numbered, 1 through 8 for the 8-puzzle, so that each tile can be uniquely identified.

The aim of the puzzle is to achieve a given configuration of tiles from a given (different) configuration by sliding the individual tiles around the grid as described above.

Searching for a Solution

This problem can be solved by searching for a solution, which is a sequence of actions (tile moves) that leads from the initial state to the goal state. Two possible states of the 8-puzzle are shown in fig 1. For search algorithms the problem is often to find the shortest solution, that is, one which consists of the least number of tile moves.

8		6
5	4	7
2	3	1

	1	2
3	4	5
6	7	8

Fig 1: Configurations of the 8-Puzzle: worst-case initial state (left) for goal state (right)

Detecting Solvable and Unsolvable 8 puzzle games

In detecting if an 8 puzzle game is solvable, you need to determine if the number of inversions in the input state is even or odd. If it is even, the puzzle is solvable, else, if it is odd, it is unsolvable.

What is inversion?

A pair of tiles form an inversion if the the values on tiles are in reverse order of their appearance in goal state.

1	2	3
4		6
8	5	7

Inversions = 3

(6-5 8-5 8-7)

Unsolvable

1	2	3
4		5
7	8	6

inversions = 2

(8-6 7-6)

Solvable

Fig. 2

What Are Heuristics?

Heuristics are rule of thumb. They are a quick way to estimate how close we are to the goal. That is, how close is a state to the goal.

For the 8-puzzle game, we can use either number of misplaced tiles or Manhattan distance to calculate the heuristic value.

Manhattan Distance

For my work, Manhattan distance will be used in calculating the heuristic value. Board configuration with the minimum Manhattan distance will be evaluated further, once Manhattan distance becomes zero (0), it means the goal state has been reached.

Manhattan distance is the sum of the absolute differences of the coordinates.

The Manhattan distance between two points is defined as:

$$D(M,P) = |M_Z - P_Z| + |M_Y - P_Y|$$

For example,

if x=(a,b) and y=(c,d), the Manhattan distance between x and y = |a-c|+|b-d|

Let’s consider two 8-puzzle board configuration below, one is the start state and the other is the goal state. the heuristics can be calculated using number of misplaced tiles or Manhattan distance.

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

Fig. 3

h1(n) = Number of misplaced tiles

h2(n) = Manhattan distance

In fig. 3 above:

h1(s) = 8

h2(s) = 3+1+2+2+2+3+3+2 = 18

Steepest Ascent Hill Climbing(Gradient Search)

It is a variant of Hill climbing algorithm. It can be implemented by having a slight modification to the simple hill climbing algorithm. In this algorithm, we consider all possible states from the current state and then pick the best one as successor, unlike in the simple hill climbing technique whereby, any state which was closer to the goal than the current state was chosen as the successor.

Steepest ascent Hill climbing algorithm

```
1   Evaluate the start state
2   If it is goal state then quit
    otherwise make the current state this start state and proceed;
3   Repeat Until a solution is found or current state does not change
        3.1  Set target to be the state that any successor of the current state can better;
        3.2  For each operator that can be applied to the current state
        3.3  Apply the new operator and create a new state
        3.4  Evaluate this state
        3.5  If this state is goal state then quit
```

3.6 Otherwise compare with target

3.7 If better set target to this value

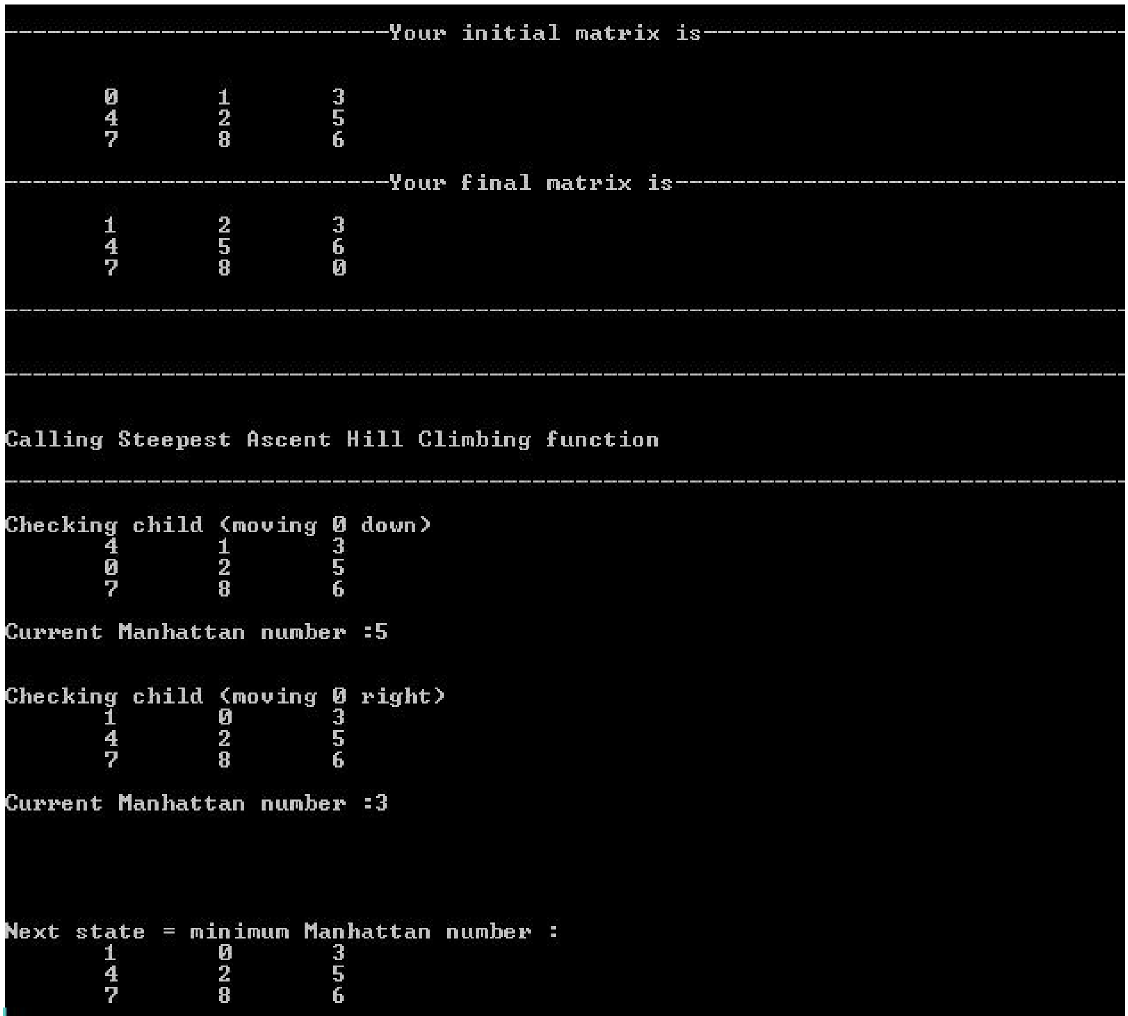
3.8 If target is better than current state set current state to target

CODE: The url to the code can be found at

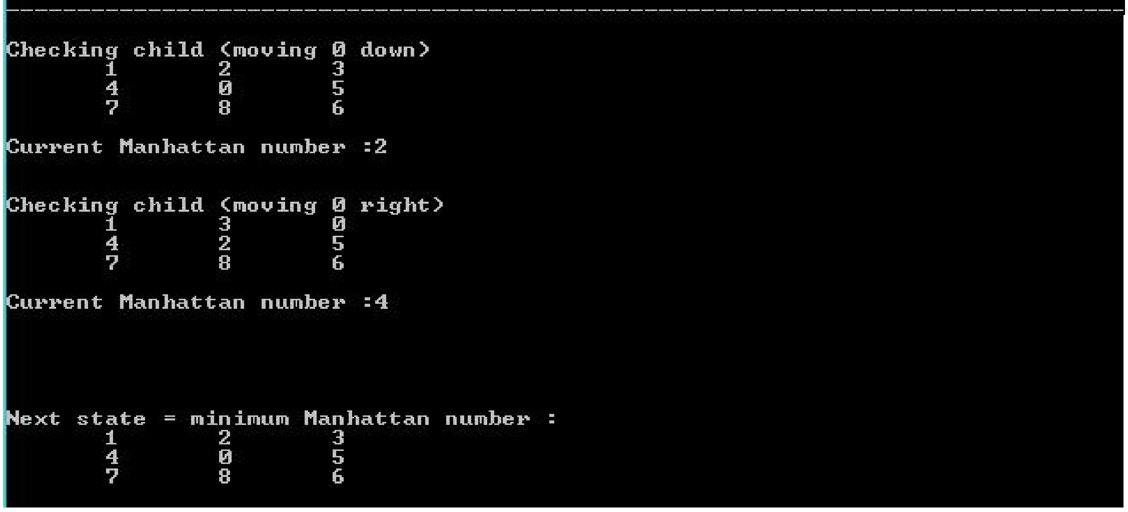
https://github.com/PatZino/8-puzzle-game-using-Steepest-Ascent-Hill-Climbing-Algorithm

The Output Is Shown Below

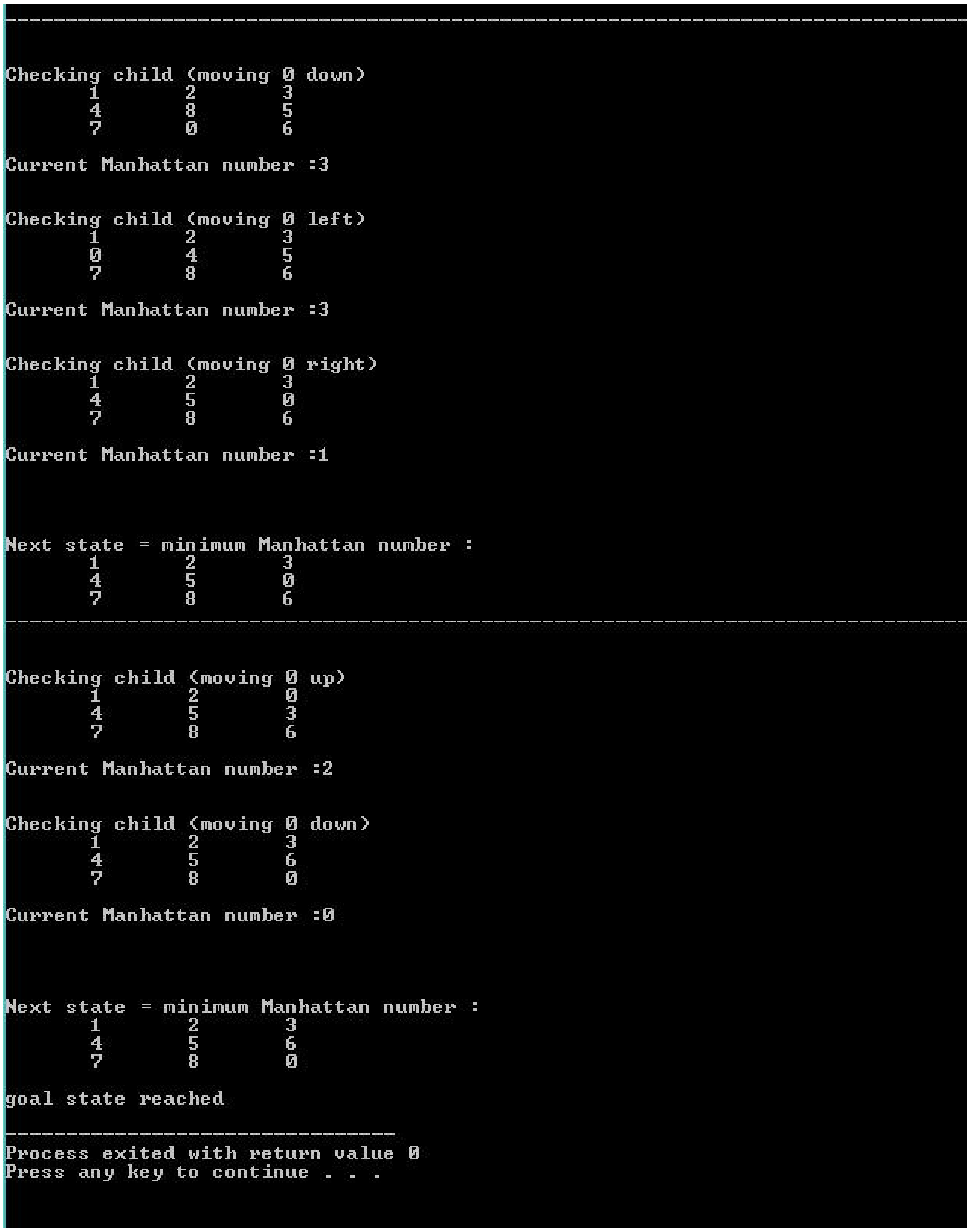
8-Puzzle using steepest ascent hill climbing algorithm -- how it operates.



```
---------------------------Your initial matrix is------------------------------

       0       1       3
       4       2       5
       7       8       6

---------------------------Your final matrix is--------------------------------

       1       2       3
       4       5       6
       7       8       0

-------------------------------------------------------------------------------


-------------------------------------------------------------------------------


Calling Steepest Ascent Hill Climbing function

-------------------------------------------------------------------------------

Checking child (moving 0 down)
       4       1       3
       0       2       5
       7       8       6

Current Manhattan number :5

Checking child (moving 0 right)
       1       0       3
       4       2       5
       7       8       6

Current Manhattan number :3




Next state = minimum Manhattan number :
       1       0       3
       4       2       5
       7       8       6
```



```
-------------------------------------------------------------------------------

Checking child (moving 0 down)
       1       2       3
       4       0       5
       7       8       6

Current Manhattan number :2

Checking child (moving 0 right)
       1       3       0
       4       2       5
       7       8       6

Current Manhattan number :4




Next state = minimum Manhattan number :
       1       2       3
       4       0       5
       7       8       6
```



```
--------------------------------------------------------------------------------


Checking child (moving 0 down)
       1       2       3
       4       8       5
       7       0       6

Current Manhattan number :3


Checking child (moving 0 left)
       1       2       3
       0       4       5
       7       8       6

Current Manhattan number :3


Checking child (moving 0 right)
       1       2       3
       4       5       0
       7       8       6

Current Manhattan number :1



Next state = minimum Manhattan number :
       1       2       3
       4       5       0
       7       8       6
--------------------------------------------------------------------------------


Checking child (moving 0 up)
       1       2       0
       4       5       3
       7       8       6

Current Manhattan number :2


Checking child (moving 0 down)
       1       2       3
       4       5       6
       7       8       0

Current Manhattan number :0



Next state = minimum Manhattan number :
       1       2       3
       4       5       6
       7       8       0

goal state reached

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

References

Geeks for Geeks. How to check if an instance of 8 puzzle is solvable? Retrieved from https://www.geeksforgeeks.org/check-instance-8-puzzle-solvable/ on 2 May 2018.

Xavier D.(2006) Manhattan distance of a point and a line. Retrieved from http://artis.imag.fr/~Xavier.Decoret/resources/maths/manhattan/html/ on 2 May 2018.

Kalev K.(2014) Informed Heuristic Search.

The 8-Puzzle. Retrieved from http://www.aiai.ed.ac.uk/~gwickler/eightpuzzle-inf.html on 2 May 2018.