PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE

ARTIFICIAL INTELLIGENCE MINI-PROJECT REPORT
ON

"USE HEURISTIC SEARCH TECHNIQUES TO IMPLEMENT HILL-CLIMBING ALGORITHM "

SUBMITTED BY

Maitraya Kakade 41427
Pranav Kulkarni 41430

Under the guidance of
Prof. Deepali Kadam

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2020-21

Contents

1 Problem Statement

Use Heuristic Search Techniques to Implement Hill-Climbing Algorithm.

2 Abstract

Heuristic functions are used to enable search algorithms make an informed guess.The idea is to guide the search so that it has tendency to explore the region leading to the goal.It is designed to help the algorithm to pick the candidate from the OPEN list that is most likely to be on the path to the goal.Hill Climbing algorithm is a modification of the best first search algorithm, where we consider the best solution in the local search space instead of the global search space. The termination criterion is that no successor of current state has better heuristic value.Here we implement the Hill climbing algorithm to solve the 8 puzzle problem.

3 Harware and Software Requirements

1. 64 bit open Source Operating System like ubuntu 18.04
2. Java-13,jdk-13
3. 500 GB HDD
4. 4GB RAM

4 INTRODUCTION

Hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution. If the change produces a better solution, another incremental change is made to the new solution, and so on until no further improvements can be found.

Hill Climbing find optimal solution for convex problems,for other it will only find local optima, which are not necessarily the best possible solution (the global optimum) out of all possible solutions (the search space).Hill climbing can often produce a better result than other algorithms when the amount of time available to perform a search is limited, such as with real-time systems, so long as a small number of increments typically converges on a good solution (the optimal solution or a close approximation).

4.1 Algorithm

Algorithm 1 Hill Climbing

```
1:  currentNode := startNode
2:  loop
3:      L := NEIGHBOURS(currentNode)
4:      nextEval := − INF
5:      nextNode := NULL
6:      for all x in L do
7:          if EVAL(x) > nextEvall then
8:              nextNode := x
9:              nextEval := EVAL(x)
10:         end if
11:     end for
12:     if nextEval <= EVAL(currentNode) then
13:         // Return current node since no better neighbors exist
14:         return currentNode
15:     end if
16:     currentNode := nextNode
17: end loop
```

4.2 State Space

4.3 Types of Hill Climbing

- Simple Hill Climbing: Simple hill climbing is the simplest way to implement a hill climbing algorithm. It only evaluates the neighbor node state at a time and selects the first one which optimizes current cost and set it as a current state. It only checks it's one successor state, and if it finds better than the current state, then move else be in the same state.

- Steepest ascent Hill Climbing:The steepest-Ascent algorithm is a variation of simple hill climbing algorithm. This algorithm examines all the neighboring

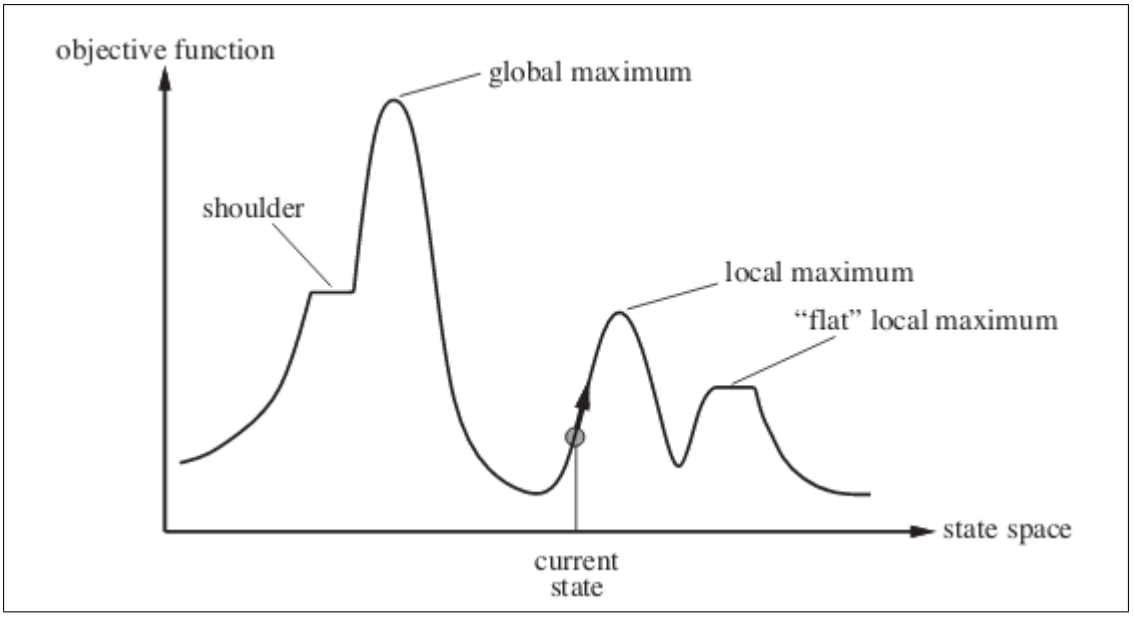

Figure 1: State Space of Hill Climbing Algorithm

nodes of the current state and selects one neighbor node which is closest to the goal state. This algorithm consumes more time as it searches for multiple neighbors.

- Stochastic Hill Climbing: It does not examine all neighbors before deciding how to move. Rather, it selects a neighbor at random, and decides (based on the amount of improvement in that neighbor) whether to move to that neighbor or to examine another.

4.4 Advantages and Disadvantages

- Advantages of Hill Climbing
  1. It uses less space than other algorithms
  2. It is very useful in job shop scheduling, automatic programming, circuit designing, and vehicle routing. is very useful in job shop scheduling, automatic programming, circuit designing, and vehicle routing.
- Disadvantages of Hill Climbing
  1. It is not complete algorithm
  2. Local Maxima: Hill climbing will not necessarily find the global maximum, but may instead converge on a local maximum.
  3. Ridges and alleys: It is a special kind of local maximum. it is an area of the search space which is higher than the surrounding areas and that itself has a slope. We can't travel the ridge by single moves as the orientation of the high region compared to the set of available moves makes it impossible.
  4. Plateau: It is a flat area of the search space in which a whole set of neighboring states(nodes) have the same order. On a plateau, it is not possible to determine the best direction in which to move by making local comparisons.

4.5 8 puzzle problem

Given a 3×3 board with 8 tiles (every tile has one number from 1 to 8) and one empty space. The objective is to place the numbers on tiles to match final configuration using the empty space. We can slide four adjacent (left, right, above and below) tiles into the empty space.

4.6 Heuristic Function

The main crux of solving this problem is the heuristic function. Here we define the heuristic function as the number of tiles that are out of place compared to goal state configuration. We define the goal state configuration as the one all tile are arranged in order from 1 to 8 with the last last tile being blank position.

5 OBJECTIVE

- To use heuristic search for hill climbing problem
- To solve 8 puzzle problem using Hill Climbing Algorithm

6 Scope

We are going to run this algorithm for 8 puzzle problem. We will not consider higher variants like 16 puzzle or 24 puzzle. Since algorithm is incomplete , we will not always find the optimal solution.We will also not consider the unsolvable instances of 8 puzzle for this exercise.

7 System Architecture

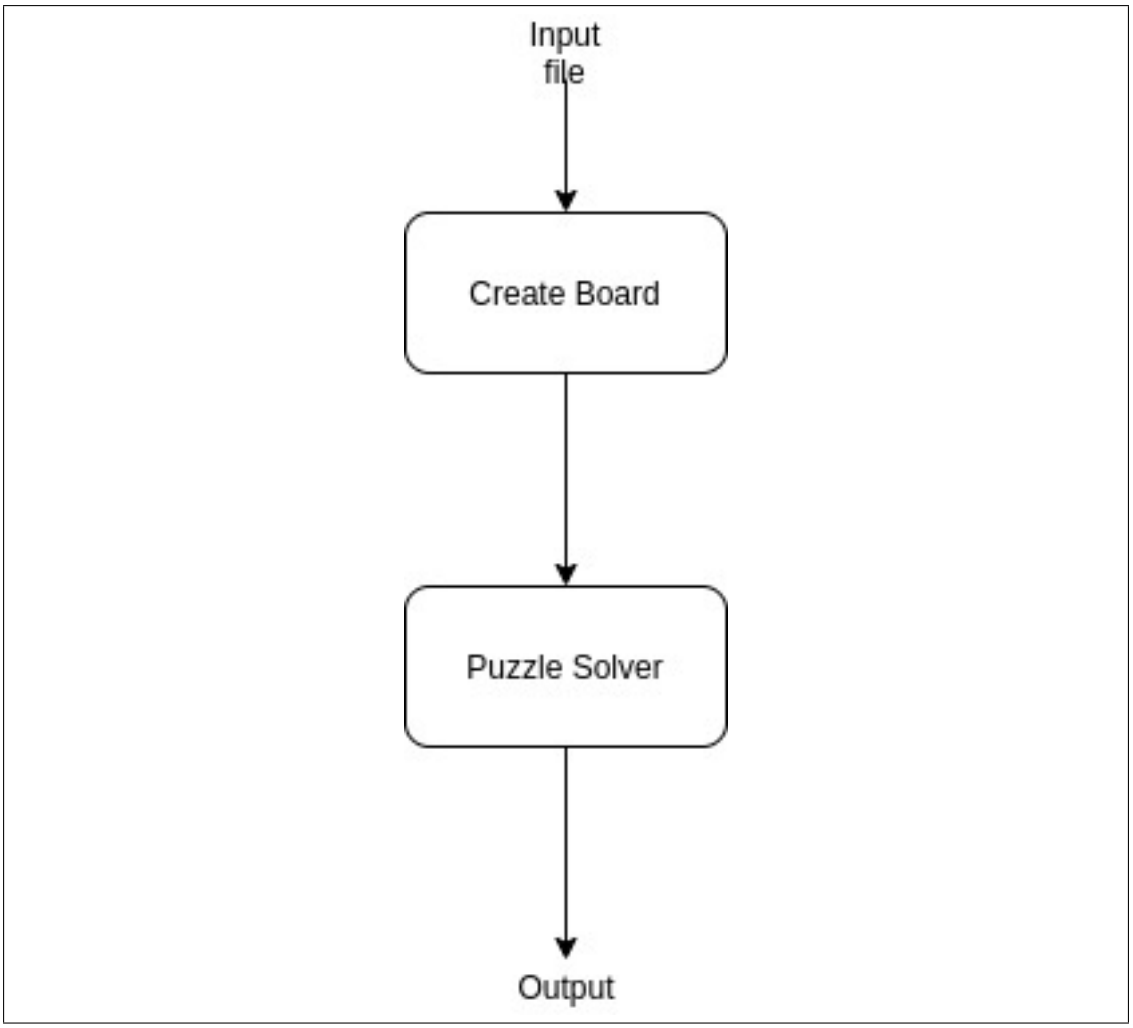

Figure 2: System Arcitecture

8 Test Cases

8.1 Test Case 1

```
Input:
3
    1  2  3
    4  5  0
    7  8  6

Output:
Minimum number of moves = 0
3
    1  2  3
    4  5  6
    7  8  0
```

8.2 Test Case 2

```
Input:
3
     1  2  3
     4  5  0
     7  8  6
Output:
Minimum number of moves = 1
3
 1  2  3
 4  5  0
 7  8  6

3
 1  2  3
 4  5  6
 7  8  0
```

8.3 Test Case 3

```
Input:
3
 1  2  3
 0  4  5
 7  8  6

Output:
Minimum number of moves = 3
3
 1  2  3
 0  4  5
```

```
 7  8  6

3
 1  2  3
 4  0  5
 7  8  6

3
 1  2  3
 4  5  0
 7  8  6

3
 1  2  3
 4  5  6
 7  8  0
```

8.4 Test Case 4(Example of finding Local optima)

```
Input:
    3
     1  3  6
     5  2  8
     4  0  7

Output:
    Minimum number of moves = 8
    3
     1  3  6
     5  2  8
     4  0  7

    3
     1  3  6
     5  0  8
     4  2  7

    3
     1  3  6
     0  5  8
     4  2  7

    3
     1  3  6
     4  5  8
     0  2  7

    3
```

```
 1  3  6
 4  5  8
 2  0  7

3
 1  3  6
 4  5  8
 2  7  0

3
 1  3  6
 4  5  0
 2  7  8

3
 1  3  0
 4  5  6
 2  7  8

3
 1  0  3
 4  5  6
 2  7  8
```

9 Result

Thus we see that the hill climbing algorithm gives the local best result when applied to solve 8 Puzzle problem. We observe how the algorithm does not always give the optimal solution.

10 Conclusion

We have studied the Hill Climbing algorithm using heuristic function. We have applied it solve 8 puzzle problem and observed that it does not always give the optimal solution.

References

[1] https://en.wikipedia.org/wiki/Hill_climbing

[2] Deepak Khemani.*A First Course in Artificial Intelligence*..1e