CS214, Lab-2 Report

Hrishikesh R. Karande (210010020) , Shivesh Pandey (210020044)

January 24, 2023

1 Introduction

The Blocks World problem involves a robot with the ability to perform operations on blocks on a table. The goal of the problem is to develop a set of rules or a program that can control the robot to stack blocks in a specific configuration.

2 Representation

2.1 State Space

The state space of a system describes the set of all possible states that the system can be in, and the transitions between these states.

2.2 Start State

Represents the starting point from which the algorithm will explore the state space in order to find a solution to the problem. The start state is typically defined by the problem's initial conditions and it can have a significant impact on the efficiency and completeness of the search, as well as the length of the solution path.

[D, C, B, A] [F,E] [ ]

2.3 Goal State

In state space search, the goal state is the final desired state that the search algorithm is trying to reach. . The search algorithm will explore different states in the state space until it reaches the goal state or determines that it is not possible to reach the goal state.

The goal state for the above mentioned example can be as [], []. [A,E,B,F,C,D]

3 Pseudo Codes

3.1 moveGen(state,heuristic)

This function takes an input as current state explores the neighbours ,applies heuristic to select the best among the neighbours.

function moveGen(state,heuristic):

```
possiblestate=[]
front=pop from i stack
            for j in 3:
                if i!=j
                    add front to stack[j]
                    add state in possiblestate
        return max(possiblestates)
    return M
```

3.2 GoalState

```
if heuristic(state)==heuristic(goalState):
    return true
```

3.3 heuristic 1

This function assigns values as +1 if present correct place else -1

```
if(xcoordinate(current)==xcoordinate(goal) and ycoordinate(current)==ycoordinate(goal))\
     h=h+1
else
     h=h-1
```

3.4 heuristic 2

This function adds values the y coordinate if present correct place else subtracts y

```
if(xcoordinate(current)==xcoordinate(goal)&&ycoordinate(current)==ycoordinate(goal))
        h=h+ycoordinate
else
        h=h-ycoordinate
```

3.5 heuristic 3

manhattan distance from the inital and goal state
for all blocks:
$h = h - |x_s - x_g| - |y_s - y_g|$

4 Running the code

- Input Format
  1. The user needs to pass a file as input which will contain initial and goal state separated by an empty line .
  2. first three lines represent the start state and the subsequent lines represent the goal state.
  3. each line contains tuple which is either empty or contains blocks with Labels.
- Output Format
  1. The code runs for the three heuristic functions defined and prints as either possible or not possible to reach.
  2. It also prints the time required for execution and the cost to reach the goal state

5 Conclusion

The Hill Climbing algorithm is a method for finding the local optimum of a function. The algorithm starts at a initial point on the function and then repeatedly moves to a neighboring point that has a higher value until it reaches a point where no neighboring point has a higher value. The conclusion that can be drawn from the Hill Climbing algorithm is that it is a simple and efficient method for finding local optima, but it may get stuck in a local optimal solution if there are multiple local optima. Additionally, it can also be sensitive to the initial starting point and may converge to a different local optima if starting from a different initial point.