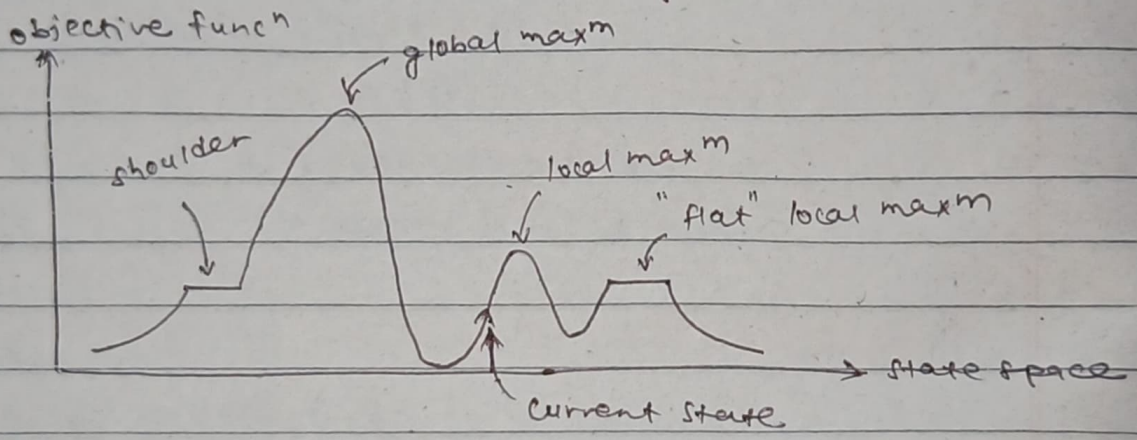
Assignment - III

1)

→ Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation value to find the peak of the mountain or best soln to the problem. It terminates when it reaches a peak value where no neighbour has a higher value.
One of the widely discussed eg of hill climbing algorithm is Travelling salesman problem in which we need to minimize the distance travelled by the salesman.

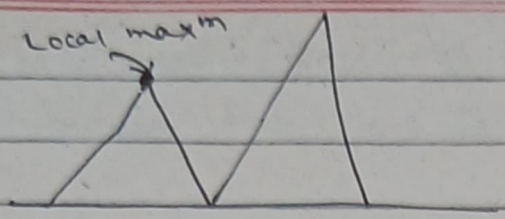


It is also called greedy local search as it only looks to its good immediate neighbour state & not beyond that. A node of hill climbing algorithm has 2 components which are state & value.

→ Problems in hill climbing:

a) Local Maxm

. It is a peak state in the landscape which is better than each of its neighbouring states, but there is another state also present which is higher than the local maxm.

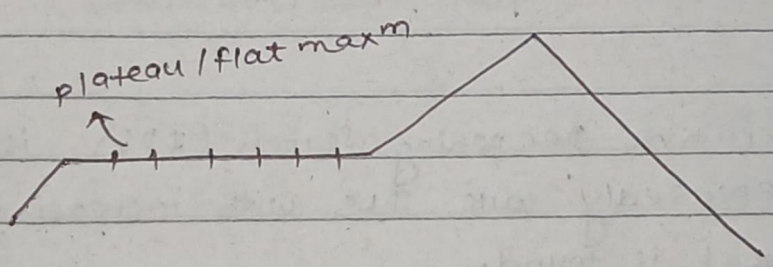


. soln

Backtracking technique can be a soln of the local maxm in state space landscape. create a list of the promising path so that the algorithm can backtrack the search space & explore other paths as well.

b) Plateau:

. It is a flat area of the search space in which all the neighbour states of the current state contains the same value, because of this algorithm doesn't find any best direction to move. A hill-climbing search might be lost in the plateau area.

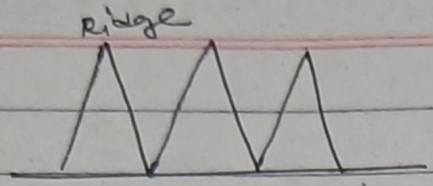


. soln

The soln for the plateau is to take big steps or very little steps while searching to solve the problem. Randomly select a state which is far away from the current state so it is possible that the algorithm could find non-plateau region.

c) Ridges:

. A ridge is a special form of the local maxm. It has an area which is higher than its surrounding areas, but



- soln

with the use of bidirectional search, or by moving in different directions, we can improve this problem.

2. Discuss DLS & IDS with example & compare these algorithms.

→ • Depth-Limited search (DLS) is a graph search algorithm that explores a graph by expanding nodes until a certain depth limit is reached.

: DLS is useful for finding soln that are guaranteed to be within a certain depth of the starting node.

• Iterative Deepening Search (IDS) is an algorithm that repeatedly calls DLS with increasing depth limits until the goal is found.

• IDS is a good choice when the depth of the soln is unknown but is likely to be small.

• Example:

If we are lost in a maze and want to find the exit. We can use DLS to explore the maze by walking down each path until we reach a dead end or a certain depth limit.

If we don't find the exit within the depth limit, we can increase the depth limit & try again.

IDS would essentially do same thing, but instead of restarting from scratch with each new depth limit, it would remember the paths it has already explored & only explore new paths.

→ Here is a comparison betn DLS & IDS:

Feature	DLS	IDS
search strategy	Depth-first	Iterative depth first
Depth limit	fixed	Increasing
Goal finding guarantee	Yes, if the goal is within the depth limit	Yes, if the goal exist & isn't too deep
space complexity	$O(d)$	$O(d)$
Time complexity	$O(b^d)$	$O(b^{d+1})$

3. Using suitable example, illustrate steps of A* search. How A* search is better than BFS?

→ A* search is the most commonly known form of best-first search. It uses heuristic function $h(n)$, and cost to reach the node n from the start state $g(n)$. It