# CS-208:Artificial Intelligence
# Lectures-06
# Informed Searching Methods

# Hill Climbing Search Technique

This algorithm searches the tree in a depth first manner, at each step choosing paths that appear to be most likely to lead to the goal.

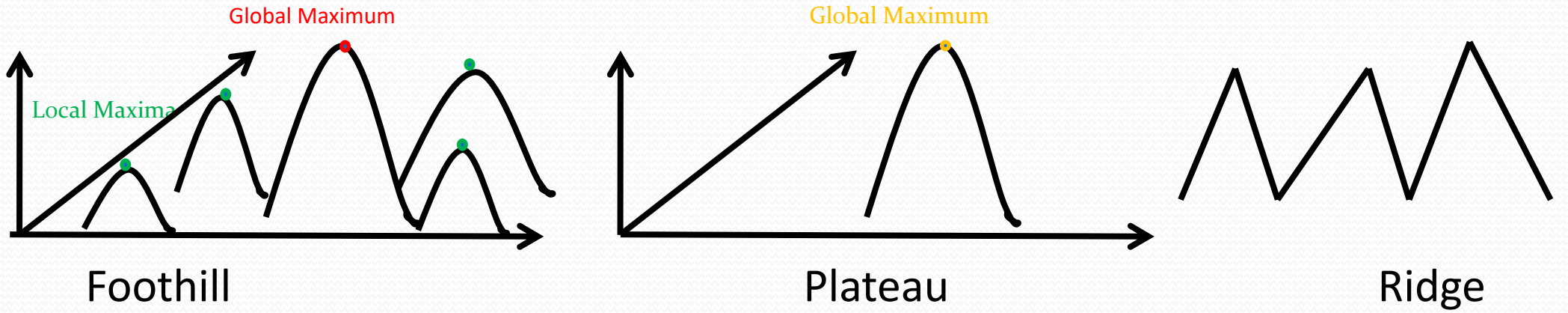
Hill climbing uses heuristic to identify paths efficiently but does not necessarily identify the best path.

Hill climbing will have difficulties in three types of search spaces namely: foothills, plateau and ridges.

**Foothills** are often called local maxima by mathematician. A local maxima is a part of the search space that appears to be preferable to the parts around it, but which is the fact just a foothill of a large hill. In other words, a local maxima is a state that is better than all its neighbours but not better than some other states farther away. The search space has a single global maximum surrounded by a number of foothills or local maxima. These local maxima draw the hill climbing procedure like magnet. An optimal point is found, but it is local not global and the user is left with false sense of accomplishment.

**A Plateau** is a region in the search space where all the values are the same. In this case, although there may well be a suitable globl maximum value somewhere nearby, there is no indication from local terrain of which direction to go to find it. Hill climbing search method could well find itself stuck in the plateau with no clear indication of where to go to find a good solution.

**A ridge** a long thin region of high land with low land on either side. A hill climbing would determine that any point on the top of the ridge was maximum because the hill falls away in all four directions. The correct direction is a very narrow one that leads to the top of the ridge, but identifying this direction using hill climbing could be very tricky.



## Hill Climbing Algorithm

Step-1: Form single element Stack consist of the root node representing starting state.

Step-2: Repeat the following until the Stack becomes empty or the goal node has been reached:

a. If first element at the front is a goal node then “do nothing”.

b. If the first element is not the goal node, then remove the first element from the queue and sort the first element successors if any with respect to estimated remaining distance to the goal node before adding them to the top of the Stack.

Step-3: If the goal node has been found then announce **SUCCESS**, otherwise **FAILURE.**

# Best First Search Technique

It is the way of combing the advantage of both depth first search and breadth first search into single method.

At each step, the best first search process will select the most promising node that have generated so far.

- ✓ expand the best partial path
- ✓ forward motion is from the best OPEN node so far and no matter where it is in the partially developed tree.
- ✓ An evaluation function is used to provide a means for order nodes on OPEN.

**Best First Search Algorithm**

Step-1: Form single element queue (called OPEN) consist of the root node representing starting state.

Step-2: Repeat the following until the queue becomes empty or the goal node has been reached:

a. If first element at the front is a goal node then “do nothing”.

b. If the first element is not the goal node, then remove the first element from the queue and add the first element successors if any to the queue. After that sort the entire queue with respect estimated remaining distance to the goal node.

Step-3: If the goal node has been found then announce **SUCCESS**, otherwise **FAILURE.**