North East University Bangladesh
Presentation
Course Code: CSE-411
Course Title: Artificial Intelligence

Presented for:
Noushad Sojib
Assistant Professor , Dept of CSE
North East University Bangladesh

Presented by:
Moriom Akter Chowdhury
Id: 180103020028
Semester:10th

Topic: Hill Climbing

Hill climbing algorithm is a local search algorithm,to find the sufficiently best solution to a problem which has a large number of possible solution. It works when a good heuristic is available.

- The algorithm is quite simple,but doesn’t find the best solution always.
- It tries to find the best solution of a problem with a random possible solution,then generate a neighbour.
- At each step the current node is replaced by the best neighbour.That means the neighbour with the highest value.
- If at some point no neighbour is better than current solution,it returns then the current solution.

Algorithm:

```
Function(problem)
      returns a state that is a local maximum
current <- MAKE-NODE(problem.INITIAL-STATE)
loop do
    neighbor<-a highest valued successor of current
if neighbor.VALUE≤ current.VALUE then
    return current.STATE
current<-neighbor
```

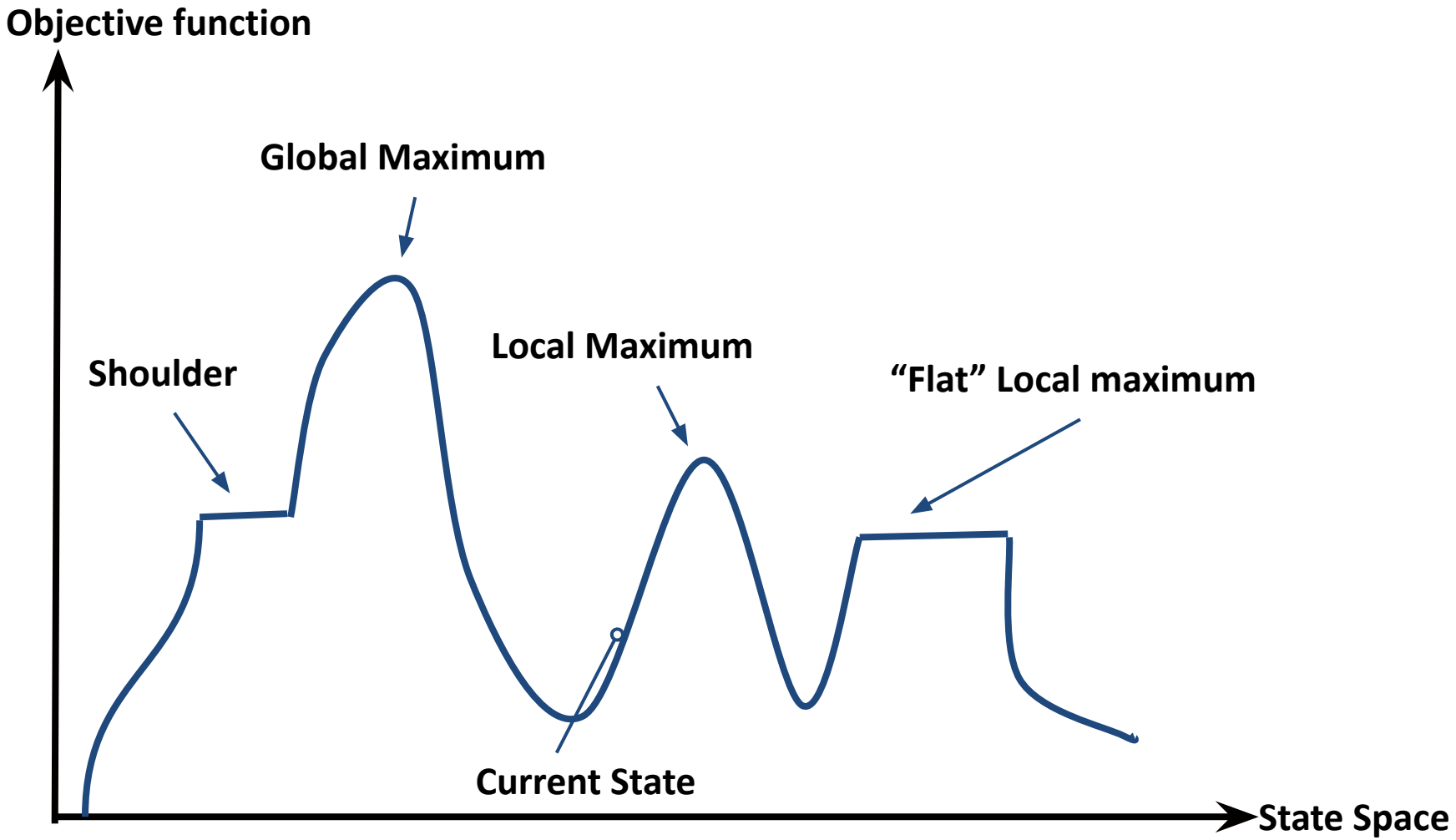
USes:

It can be used in travelling salesmen,8-qween problem and some other problems.

In which we want an optimal/minimize solution.

Diagram:

>


Objective function
Global Maximum
Shoulder
Local Maximum
“Flat” Local maximum
Current State
State Space

Limitations:

local maximum: It can get stuck in a local maximum where non of the direct neighbour of the current solution is better then the current solution.Though gets drawn towards the peak and stuck there,no way to go.

Ridges: These are sequences of the local maxima.

Plateaux: This is a “flat” state space region.As there is no uphill to go,algorithm gets stuck.

Advantages:

We don't need to maintain any search tree or graph,though it has only a single current state.

Disadvantages:

This algorithm is neither complete nor optimal.

Agent & Environment

Goal of agent:
High performance
optimized result
Rational action
PEAS: Performance,Environment,Action,Sensor

Agent-Percept-Decision-Action

Types:
simple reflex agent
model based reflex
Goal based agent
utility based agent
learning agent

Simple Reflex Agent

- Act only on the basis of current perception
- Ignore the rest of percept history
- Based on if-then rules
- Environment should be fully observable

Environment-> percept(sense) ->current situation ->if then-> action

Model Based Agent

- Partially observable environment
- Store percept history(Internal model)

Goal Based Agent

- Expansion of model-based reflex agent
- Desirable situation(Goal)
- Searching & planning

Utility Based Agent

- Partially observable environment
- Utility function
- Deals with happy and unhappy function
- Focus on Utility not goal

Genetic algorithm(John Holland)

- Abstraction of real biological evolution
- Solve complex problem(like NP hard)
- Focus on optimization
- Population of possible solution for a given problem
- From a group of individuals the best will survive