Local Search Algorithms

Hill Climbing, Simulated Annealing

Local Search Algorithms

- ❖ Local search can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions.
- ❖ Local search algorithms move from solution to solution in the space of candidate solutions (the search space) until a solution deemed optimal is found or a time bound is elapsed.
- ❖ For example, the travelling salesman problem, in which a solution is a cycle containing all nodes of the graph and the target is to minimize the total length of the cycle. i.e. a solution can be a cycle and the criterion to maximize is a combination of the number of nodes and the length of the cycle.

Local Search Algorithms

- ❖ Terminate on a time bound or if the situation is not improved after number of steps.
- ❖ Local search algorithms are typically incomplete algorithms, as the search may stop even if the best solution found by the algorithm is not optimal.
- ❖ A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution.

Local Search Algorithms

- ❖ Local search problem have three main entities:
- ❖ **Search Space:** It is a typical representation in the form of a graph, where nodes represent partial or complete solution of the problem and each of the arcs corresponds to the problem-solving process.
- ❖ **Neighborhood Relations:** In general a neighborhood of a point without leaving the region.
- ❖ **Cost function:** The selection of the move to be performed at each step of the search is based on the cost function.

Hill Climbing Search

- ❖ 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 solution to the problem.
- ❖ It terminates when it reaches a peak value where no neighbor has a higher value.
- ❖ Hill climbing algorithm is a technique which is used for optimizing the mathematical problems.
- ❖ One of the widely discussed examples of Hill climbing algorithm is Traveling-salesman Problem in which we need to minimize the distance traveled by the salesman.

Hill Climbing Search

- ❖ It is also called greedy local search as it only looks to its good immediate neighbor state and not beyond that.
- ❖ A node of hill climbing algorithm has two components which are state and value.
- ❖ Hill Climbing is mostly used when a good heuristic is available.
- ❖ In this algorithm, we don't need to maintain and handle the search tree or graph as it only keeps a single current state

Features of Hill Climbing

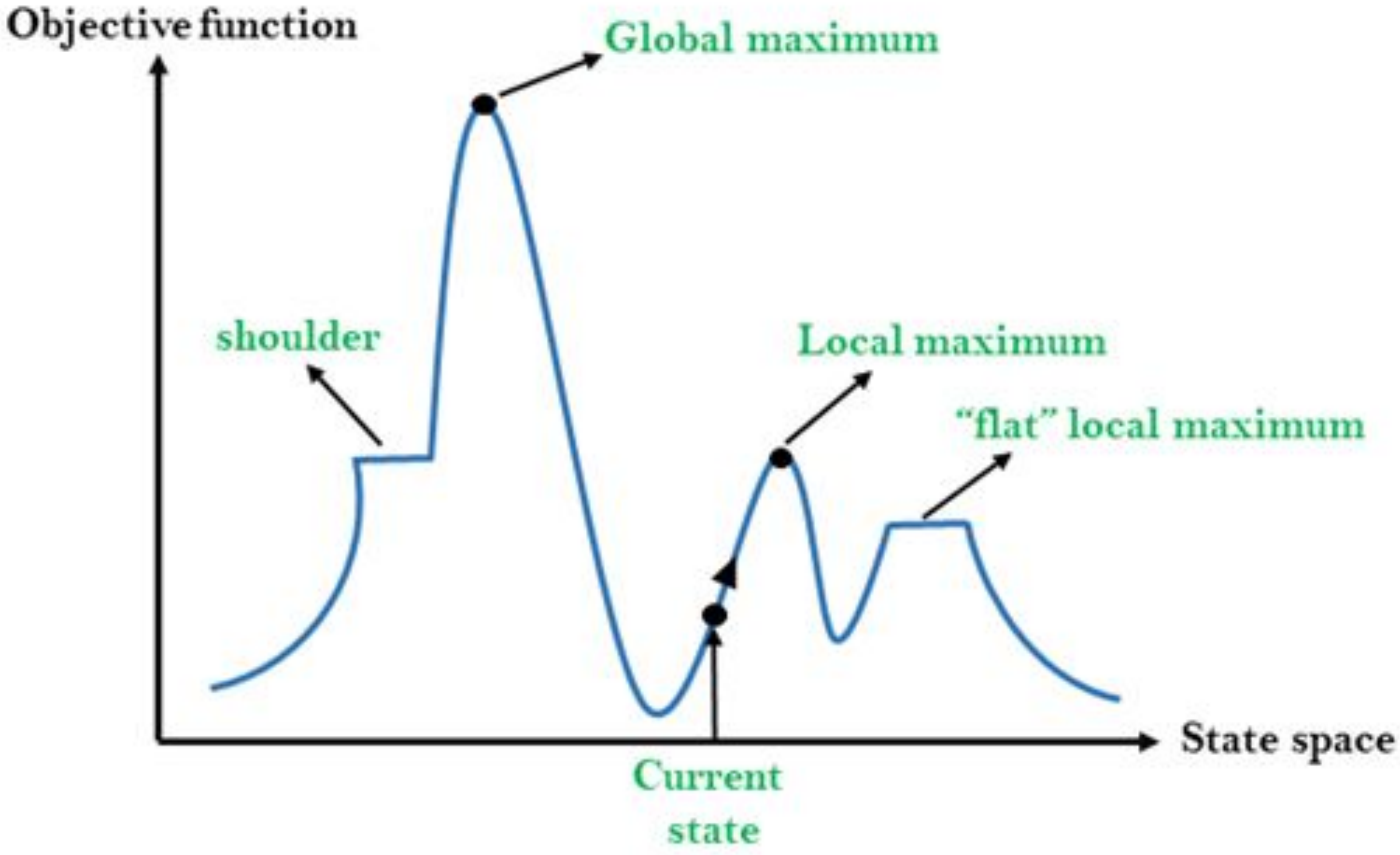
- Following are some main features of Hill Climbing Algorithm:
  - **Generate and Test variant:** Hill Climbing is the variant of Generate and Test method.

    The Generate and Test method produce feedback which helps to decide which direction to move in the search space.
  - **Greedy approach:** Hill-climbing algorithm search moves in the direction which optimizes the cost.
  - **No backtracking:** It does not backtrack the search space, as it does not remember the previous states.

State-space Diagram for Hill Climbing

- ❖ The state-space landscape is a graphical representation of the hill-climbing algorithm which is showing a graph between various states of algorithm and Objective function/Cost.
- ❖ On Y-axis we have taken the function which can be an objective function or cost function, and state-space on the x-axis.
- ❖ If the function on Y-axis is cost then, the goal of search is to find the global minimum and local minimum.
- ❖ If the function of Y-axis is Objective function, then the goal of the search is to find the global maximum and local maximum.

State-space Diagram for Hill Climbing



Different regions in the state space landscape

- **Local Maximum:** Local maximum is a state which is better than its neighbor states, but there is also another state which is higher than it.
- **Global Maximum:** Global maximum is the best possible state of state space landscape. It has the highest value of objective function.
- **Current state:** It is a state in a landscape diagram where an agent is currently present.
- **Flat local maximum:** It is a flat space in the landscape where all the neighbor states of current states have the same value.
- **Shoulder:** It is a plateau region which has an uphill edge.

Types of Hill Climbing Algorithm

- Types of Hill Climbing Algorithm:
  1. Simple hill Climbing
  2. Steepest-Ascent hill-climbing
  3. Stochastic 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.
- This algorithm has the following features:
  1. Less time consuming
  2. Less optimal solution and the solution is not guaranteed

Simple Hill Climbing

- ❖ Algorithm for Simple Hill Climbing:

Step 1: Evaluate the initial state, if it is goal state then return success and Stop.

Step 2: Loop Until a solution is found or there is no new operator left to apply.

Step 3: Select and apply an operator to the current state.

Step 4: Check new state:

  i. If it is goal state, then return success and quit.
  ii. Else if it is better than the current state then assign new state as a current state.
  iii. Else if not better than the current state, then return to step2.

Step 5: Exit.

Steepest-Ascent hill climbing

- The steepest-Ascent algorithm is a variation of simple hill climbing algorithm.
- This algorithm examines all the neighboring 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

Steepest-Ascent hill climbing

Algorithm for Steepest-Ascent hill climbing:

Step 1: Evaluate the initial state, if it is goal state then return success and stop, else make current state as initial state.

Step 2: Loop until a solution is found or the current state does not change.

i. Let SUCC be a state such that any successor of the current state will be better than it.

ii. For each operator that applies to the current state:

   a) Apply the new operator and generate a new state.

Steepest-Ascent hill climbing

a) Apply the new operator and generate a new state

b) Evaluate the new state.

c) If it is goal state, then return it and quit, else compare it to the SUCC.

d) If it is better than SUCC, then set new state as SUCC.

e) If the SUCC is better than the current state, then set current state to SUCC.

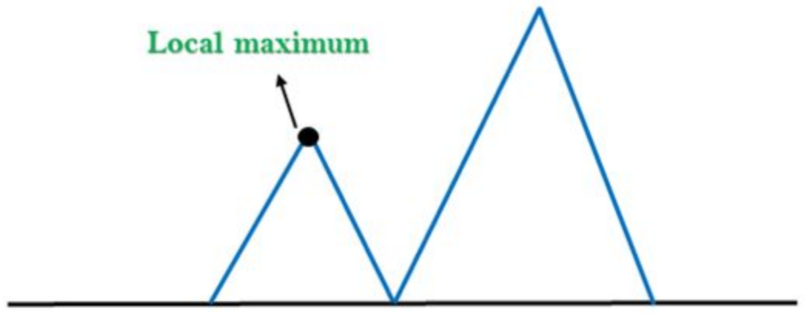
Step 5: Exit.

Stochastic hill climbing

- ❖ Stochastic hill climbing does not examine for all its neighbor before moving.
- ❖ Rather, this search algorithm selects one neighbor node at random and decides whether to choose it as a current state or examine another state.

Problems in Hill Climbing Algorithm

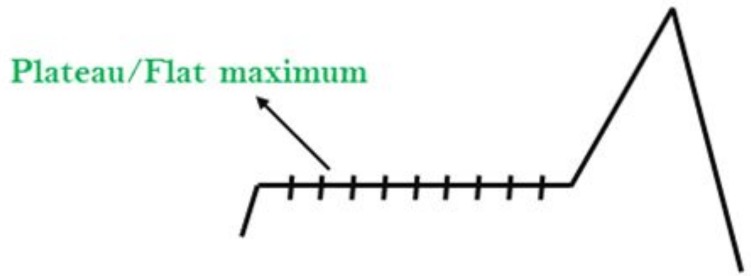
- **Local Maximum:** A local maximum is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.



- **Solution:** Backtracking technique can be a solution of the local maximum in state space landscape. Create a list of the promising path so that the algorithm can backtrack the search space and explore other paths as well.

Problems in Hill Climbing Algorithm

- **Plateau:** A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorithm does not find any best direction to move. A hill-climbing search might be lost in the plateau area.



- **Solution:** The solution 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.

Problems in Hill Climbing Algorithm

- **Ridges:** A ridge is a special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope, and cannot be reached in a single move.

- **Solution:** With the use of bidirectional search, or by moving in different directions, we can improve this problem.

Simulated Annealing

- A hill-climbing algorithm which never makes a move towards a lower value guaranteed to be incomplete because it can get stuck on a local maximum.
- If algorithm applies a random walk, by moving a successor, then it may complete but not efficient.
- Simulated Annealing is an algorithm which yields both efficiency and completeness.

Simulated Annealing

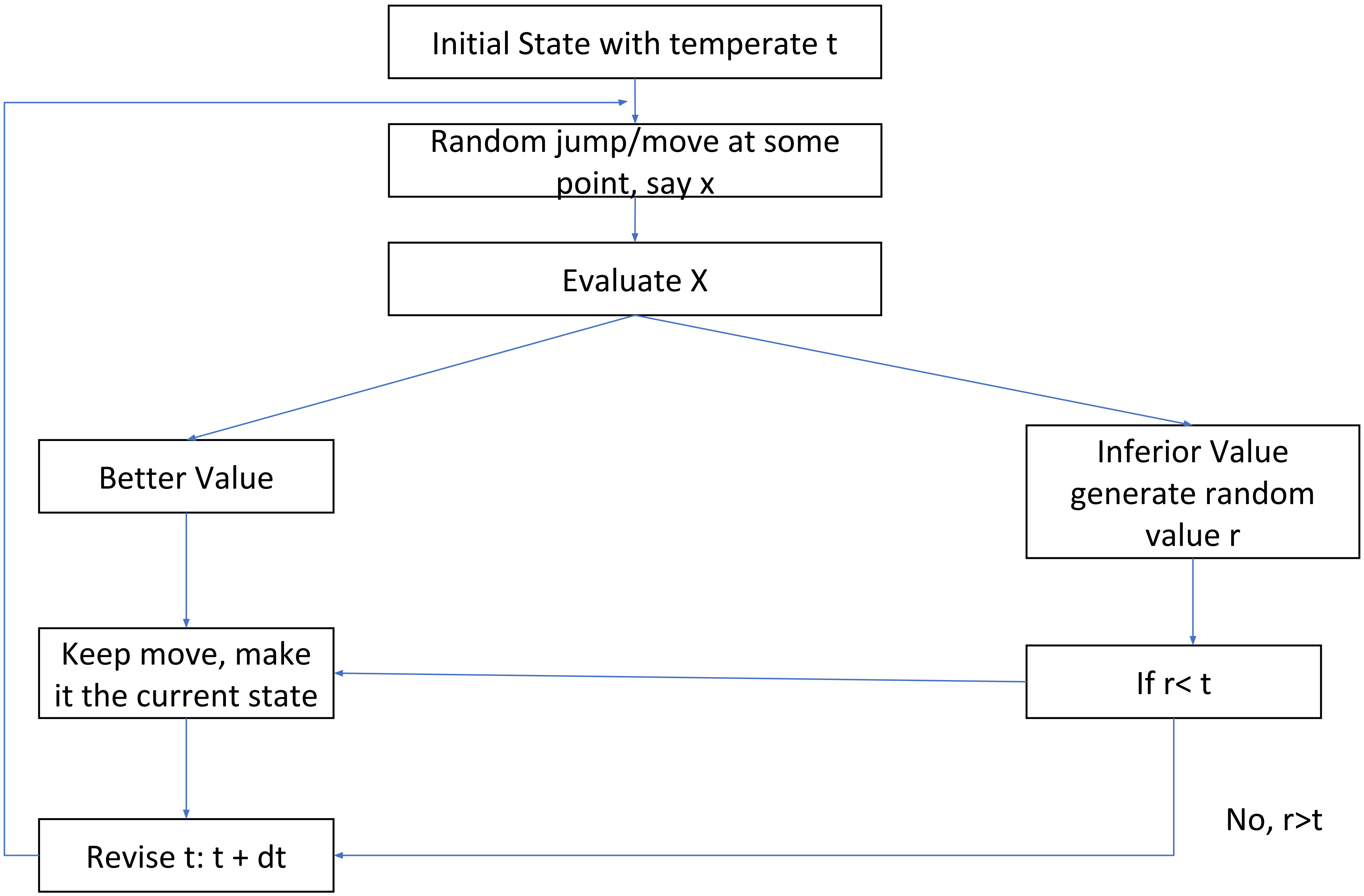
- In mechanical term Annealing is a process of hardening a metal or glass to a high temperature then cooling gradually, so this allows the metal to reach a low-energy crystalline state.
- The same process is used in simulated annealing in which the algorithm picks a random move, instead of picking the best move.
- If the random move improves the state, then it follows the same path.
- Otherwise, the algorithm follows the path which has a probability of less than 1 or it moves downhill and chooses another path.

Simulated Annealing

- Motivated by the physical annealing process
- Material is heated and slowly cooled into a uniform structure
- Simulated annealing mimics this process
- The first SA algorithm was developed in 1953 (Metropolis)

Simulated Annealing

- Compared to hill climbing the main difference is that SA allows downwards steps
- Simulated annealing also differs from hill climbing in that a move is selected at random and then decides whether to accept it
- In SA better moves are always accepted. Worse moves are not



Simulated Annealing: Basic Steps

Algorithm for Simulated Annealing

1. Let IS be the initial state and GS be the goal state
2. Check if IS = GS. If true, return success

   Else

   i. Initialize t for IS, where t is temperature or energy level to allow maximum movement
   ii. Do while a stop condition is not satisfied.
      a) Randomly pic a neighbour, say x
      b) Evaluate x.
      c) ΔE =Energy_val(x) – Energy-Val(IS)
      d) If x = GS, Return success
      e) If $\Delta E<0$, make x to be IS. (This means new state/solution is a better one)
      f) Else generate random number: r[0,1].
      g) If $r < e^{-\Delta E/t}$, make x to be IS
      h) Revise t values.