**Hill-Climbing Search**

It is an iterative algorithm that starts with an arbitrary solution to a problem and attempts to find a better solution by changing a single element of the solution incrementally. If the change produces a better solution, an incremental change is taken as a new solution. This process is repeated until there are no further improvements.

function Hill-Climbing (problem), returns a state that is a local maximum.

inputs: problem, a problem

local variables: current, a node

neighbor, a node

current <-Make\_Node(Initial-State[problem])

loop

do neighbor <- a highest\_valued successor of *current*

if Value[neighbor] ≤ Value[current] then

return State[current]

current <- neighbor

end

**Disadvantage** − This algorithm is neither complete, nor optimal.

### Simulated Annealing

Annealing is the process of heating and cooling a metal to change its internal structure for modifying its physical properties. When the metal cools, its new structure is seized, and the metal retains its newly obtained properties. In simulated annealing process, the temperature is kept variable.

We initially set the temperature high and then allow it to ‘cool' slowly as the algorithm proceeds. When the temperature is high, the algorithm is allowed to accept worse solutions with high frequency.

Start

* Initialize k = 0; L = integer number of variables;
* From i → j, search the performance difference ∆.
* If ∆ <= 0 then accept else if exp(-Δ/T(k)) > random(0,1) then accept;
* Repeat steps 1 and 2 for L(k) steps.
* k = k + 1;

Repeat steps 1 through 4 till the criteria is met.

End