AI PROJECT

Application of Simulated Annealing To Optimization Problems

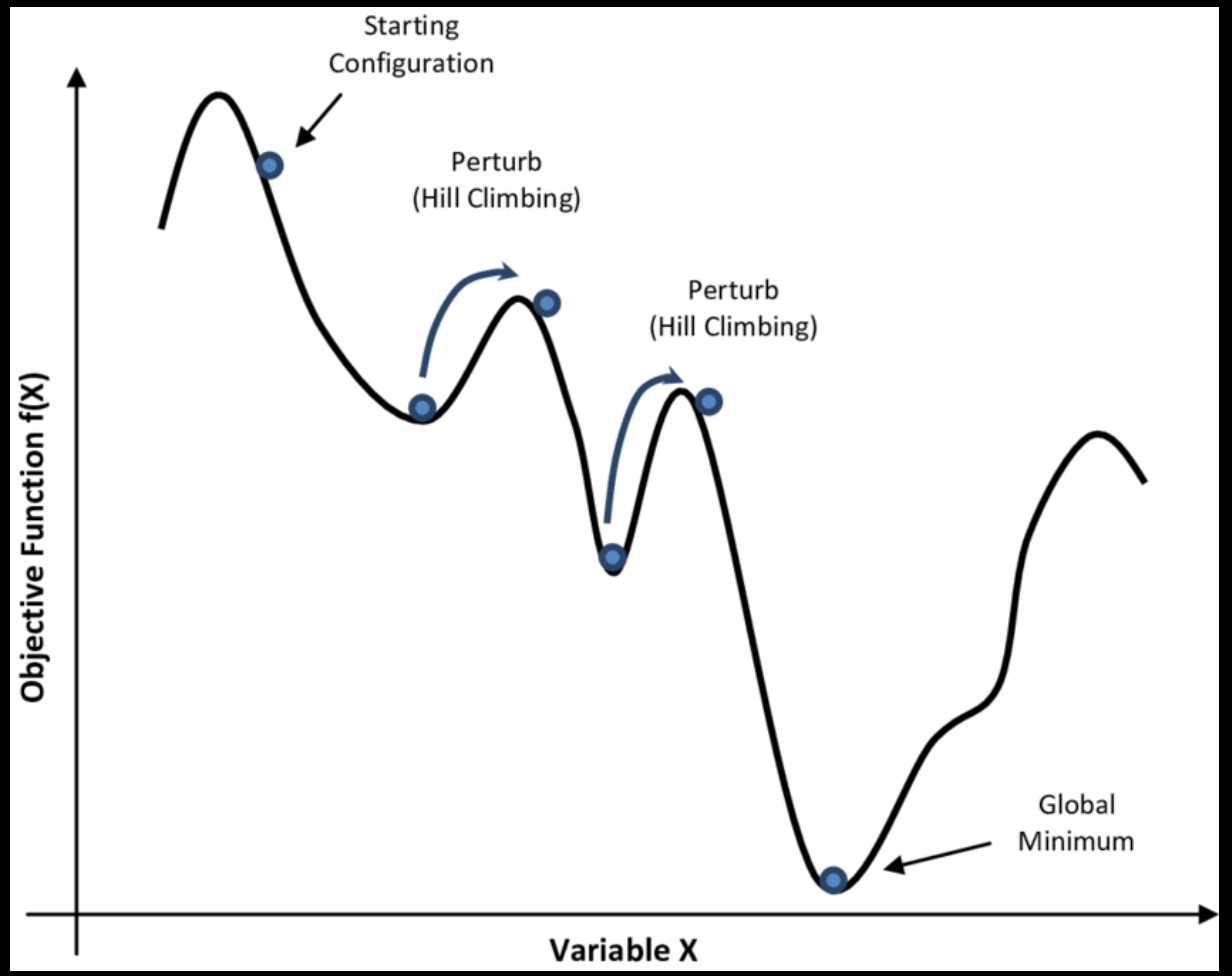
Name – Hrishikesh Athalye

MIS – 111803154

Class – TY Comp Div 2

SIMULATED ANNEALING - OVERVIEW

- Simulated Annealing is an algorithm in the category of *metaheuristic algorithms* – algorithms that help to choose a particular heuristic.
- Metaheuristic algorithms are often inspired from natural phenomenon and they often yield approximately optimal results and an be used to *reduce running time on NP Hard problems.*
- Simulated Annealing is inspired from the metallurgical principle of annealing – Controlled cooling of a heated material to produce desired physical characteristics.
- In implementation, it is a mixture of *exploration and exploitation* strategies and helps *avoid local optima,* something that simple hill climbing fails to achieve. It does so by also accepting bad moves with some probability that changes with temperature.



IMPLEMENTATION

- All the three problems taken by me use the pseudocode on the right.
- The commonly implemented version of the algorithm includes two loops:
  - A. An outer loop that checks if the temperature has fallen below a desired value
  - B. An inner loop that decides number of iterations in each temperature.
- The temperature is decreased using a *cooling schedule* which controls how fast the temperature drops.
- The number of iterations per temperature is also a function of temperature depending on if more iterations are needed at lower or higher temperatures.
- The algorithm is a modification to stochastic hill climbing. *The agent generates a random neighbour* and if its cost is less it is accepted immediately. If its cost is higher then also it can be accepted, but with some probability – decided by computing the sigmoid function on the difference in costs.
- At high temp, this probability is also high which means that the algorithm *focuses on exploration initially* , makes more mistakes and at lower temperatures, near stopping it does not reward bad moves and only focuses on exploitation of the metric.
- This strategy makes it better than hill climbing and stops it from getting stuck in local optima.

```
procedure simulated annealing
begin
   t ← 0
   initialize T
   select a current point v_c at random
   evaluate v_c
   repeat
      repeat
         select a new point v_n
            in the neighborhood of v_c
         if eval(v_c) < eval(v_n)
            then v_c ← v_n
            else if random[0,1) < e^((eval(v_n)−eval(v_c))/T)
                 then v_c ← v_n
      until (termination-condition)
      T ← g(T,t)
      t ← t + 1
   until (halting-criterion)
end
```

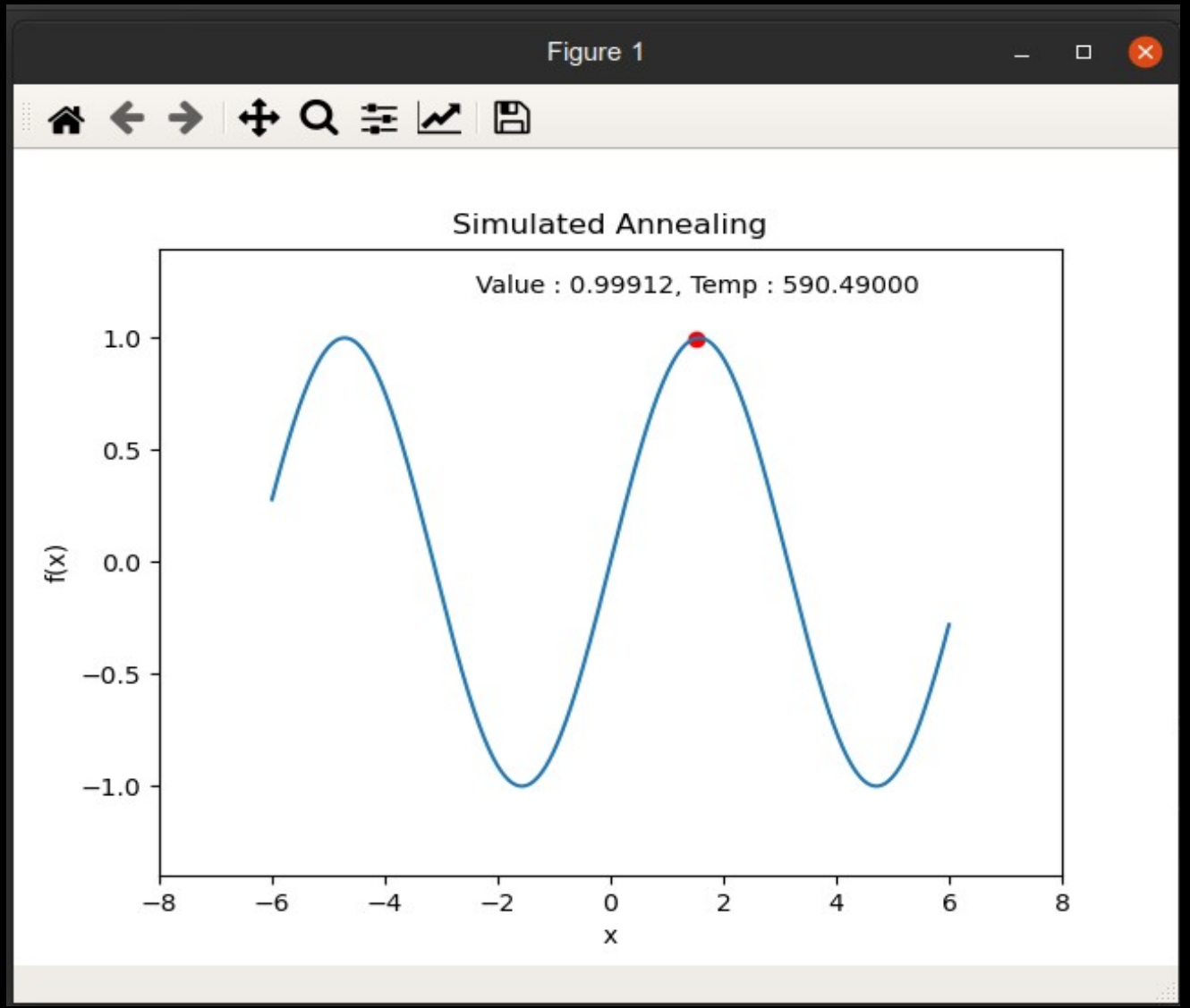
Fig. 5.3. Structure of simulated annealing

- Taken from How to Solve It: Modern Heuristics – By Michalewicz

PROBLEMS SOLVED AND AGENTS

Optima Finding Agent – An agent that finds the optima in a function.

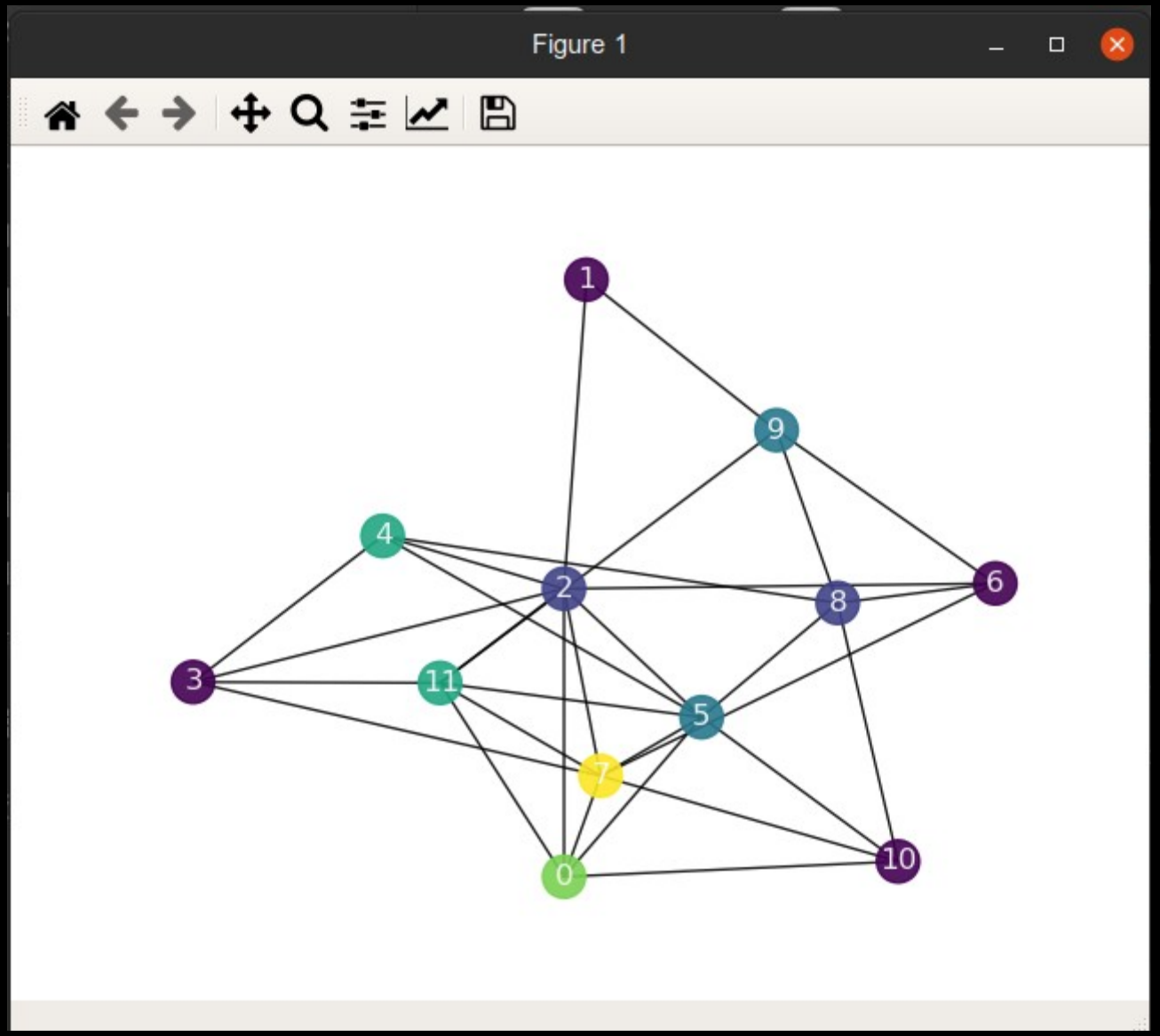
- Heuristic - SA + Greedy Heuristic.
- Search Space – 1d Function Curve
- Goal – Minimum/Maximum f(x)



PROBLEMS SOLVED AND AGENTS

Map Coloring Problem – NP Hard

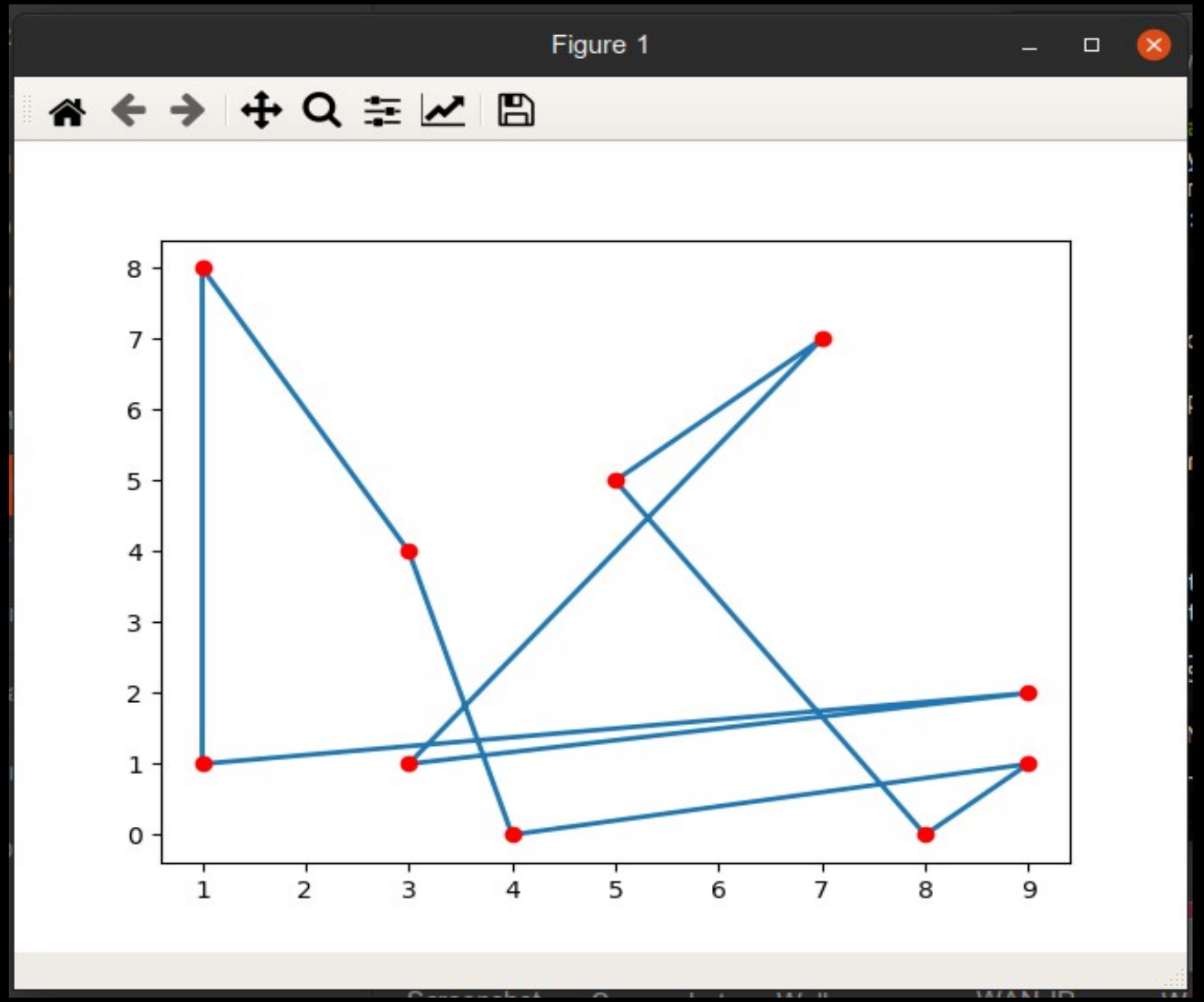
- Heuristic – SA + Welsh Powell Heuristic
- Search Space – Nodes of the graph
- Goal – Colour all nodes of the map such that no two adjacent nodes have same colour, minimising number of colours



PROBLEMS SOLVED AND AGENTS – CONTD.

Travelling Salesman Problem – NP hard

- Agent – Travelling Salesman
- Heuristic – SA + 2-Opt
- Search Space – Graph of cities
- Goal – Reach back to start node by visiting all nodes exactly once, minimising path cost.



FURTHER IMPROVEMENTS

- Choosing different cooling schedules – arithmetic, geometric, etc.
- Choosing appropriate number of inner loop iterations.
- Pairing with other local search strategies like random jumps, remembering history.
- Combining with better heuristics.