Simulated Annealing

Author: Gur Saran Adhar

1 Introduction

Physical annealing is the process of heating a metal until it melts, then cooling it slowly. The purpose of physical annealing is to produce a strong, defect-free crystal with regular structure. When the material is hot, the atoms are in a higher energy state and rearrange themselves more easily. As the temperature drops, the atomic energy decreases, and the atoms do not rearrange as easily. Slow cooling allows the material to reach a state of minimum energy, which is its crystalline form.

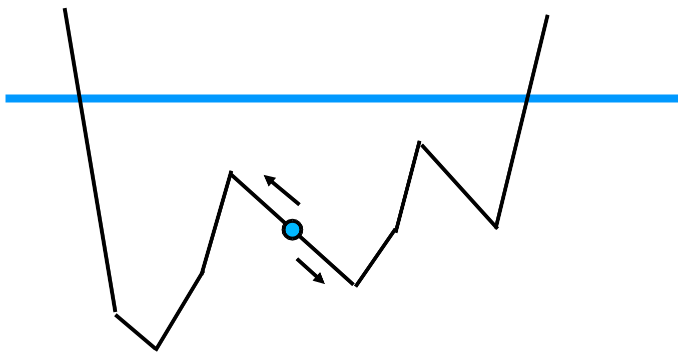
Simulated annealing makes an analogy between physical annealing and solving an optimization problem. A solution to the optimization problem corresponds to a state of the material, the value of the objective function for a particular solution corresponds to the energy associated with a particular state, and the optimal solution to the problem corresponds to the minimum energy state.

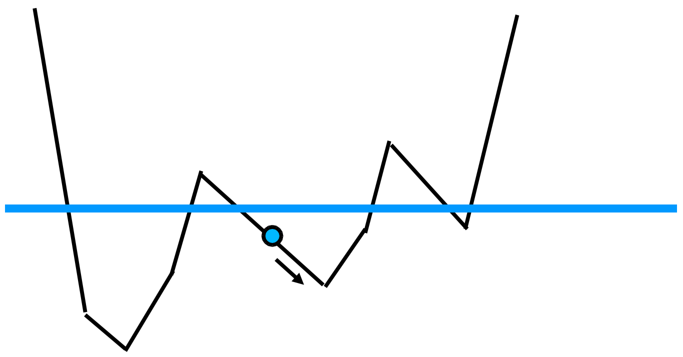
Simulated annealing is an iterative algorithm. During each iteration the current solution is *randomly changed* to create an alternative solution in the neighborhood of the current solution. **If the value of the objective function for the new solution is less than the value of the objective function for the current solution, then the new solution becomes the current solution. If the value of the objective function for the new solution is greater than the value of the objective function for the current solution, then the new solution becomes the current solution with probability $e^{-\Delta/T}$, where Δ is the difference between the values of the objective functions and T is the current "temperature".**

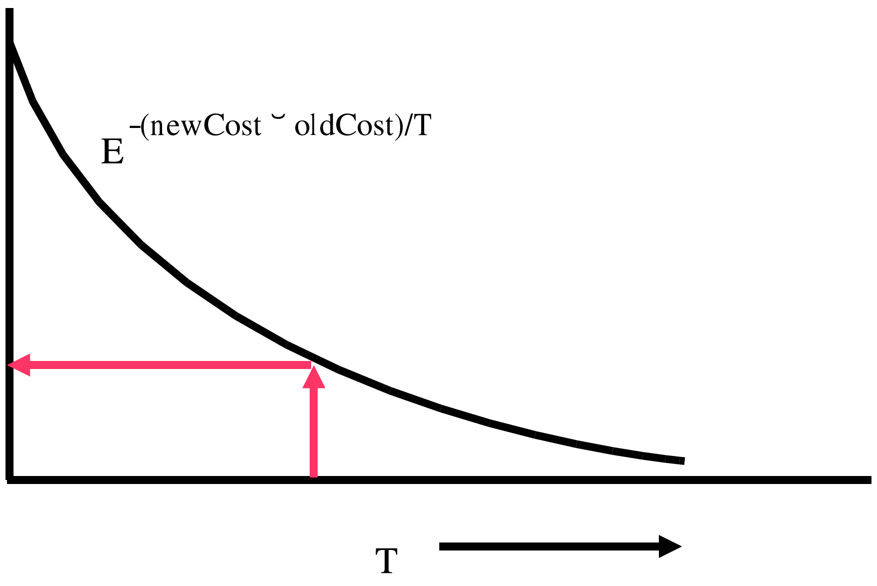
Why would we want to move to a solution that is inferior to one we have already found? The reason is that the solution space usually have local minima. We don't want the algorithm to settle too quickly into a local minimum. When the temperature is higher, the algorithm can easily "climb out of" local minima. When the temperature decreases, the probability of doing so is reduced (refer to the following figure).

2 TSP- The Traveling Salesman Problem

A (sequential) C program which randomly generates coordinates of points (cities) and computes the shortest tour is used to illustrate simulated an-

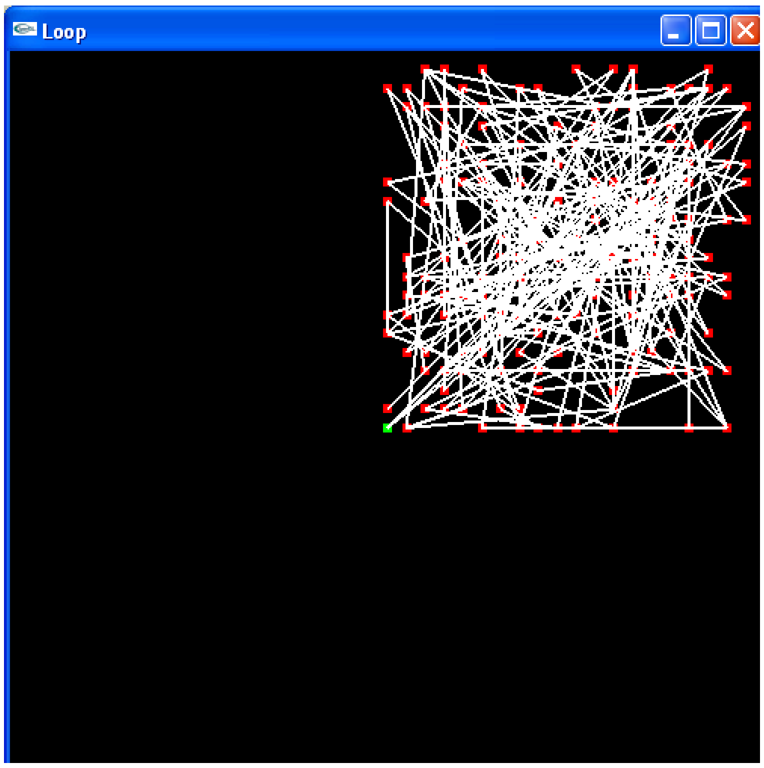





E^{-(newCost ˘ oldCost)/T}
T

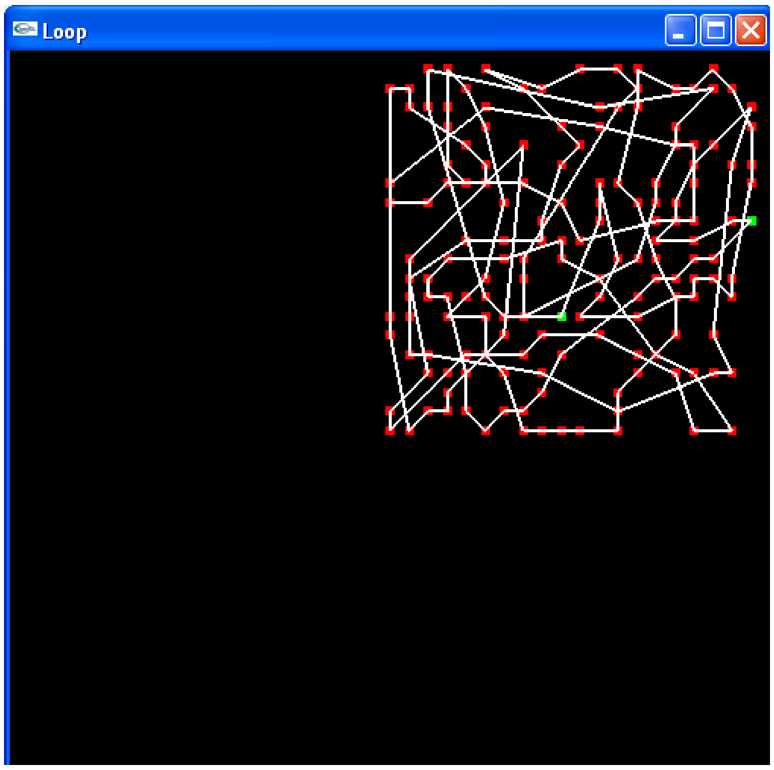
nealing. This program uses the OpenGL and the GLUT for visualizing the progress of the algorithm. A single click on the left mouse button performs one iteration of the annealing whereas the right mouse click moves forward the annealing process uninterrupted. A click on the middle mouse suspends the annealing process.

A screen shot showing the initial configuration of the problem with 200 points is shown below.



Simulated annealing is not guaranteed to find optimal solution. In fact, the same algorithm using different streams of random numbers may converge on to different solutions. Hence it makes sense to execute the same algorithm multiple times using different random number seeds. A second screen shot after the algorithm has progressed somewhat shows clear improvement.

It should be emphasized that the choice of cooling function can have much effect on the performance of the algorithm. A poor function may cause a simulated annealing algorithm to find a poor solution. In this implementation we are using a very simple geometric cooling function:



$$T_{i+1} = 0.999 * T_i$$