11/23/2013

Markov Chain Monte Carlo and Its application for Travelling Salesman Problem

Nikunj Amipara (201001199)

Team members
Nikunj Amipara(201001199)
Parth Shah (201001200)
Om Thakkar(201001203)

1. Theory

1.1 Markov Chains:

Markov chains are random sequence of finite states and the transition probability for switching states is totally depends on the current states only.

Let us suppose we have a set of states, $S = \{s_1, s_2, \ldots . s_r\}$. The process starts in one of these states and moves successively from one state to another. Each move is called a step. If the chain is currently in state s_i, then it moves to state s_j at the next step with a probability denoted by p_{ij} , and this probability does not depend upon which states the chain was in before the current state.

The Probabilities are called Transition Probabilities and the matrix containing all such probabilities is called Transition Matrix. The process can remain in the state it is in, and this occurs with probability p_{ii}. An initial probability distribution, defined on S, specifies the starting state. Usually this is done by specifying a particular state as the starting state.

A sequence {Xn: n>=0} of random variables taking values in a countable state space S is called a Marcov Chain if,

$$P\{Xn = \alpha n | Xn-1=\alpha n-1, Xn-2=\alpha n-2 \ldots\ldots X0=\alpha 0\} = P\{Xn=\alpha n \mid Xn-1=\alpha n-1\}$$

I.e. the probability that the chain will be in a certain state αn at time n given all its past history depends only on its previous state at time n-1.

Usually we also impose the condition of homogeneity

$$P\{Xn+1=j \mid Xn=i\} = P\{X1=j \mid X0=i\}$$

1.2 Markov Chain Monte Carlo:

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. It is generally used when the random vector to be simulated is composed of a large number of independent variables to generate Markov chain whose equilibrium distribution is equal to required distribution.

1.3 Simulated Annealing:

Simulated Annealing is a generic probabilistic meta-algorithm used to find an approximate solution to global optimization problems. It is inspired by annealing in metallurgy which is a technique of controlled cooling of material to reduce defects.

The simulated annealing algorithm takes random walks through the problem space. It looks for the points with low energies; in these random walks, the probability of taking a step is determined by the Boltzmann distribution,

$$p = e^{-(E_{i+1}-E_i)/T} \; if \; E_{i+1} > E_i$$

$$p = 1 \qquad\qquad if \; E_{i+1} < E_i$$

Here p is probability of taking next step, E_i is energy of current step. E_{i+1} is energy of next step .

In other words, a step will occur if the new energy is lower. If the new energy is higher, the transition can still occur with some probability which is proportional to the temperature T and inversely proportional to the energy difference $E_{i+1} - E_i$. The temperature T is initially set to a high value, and a random walk is carried out at that temperature. Then the temperature is lowered very slightly according to a cooling rate c, for an example: T changes to $T * (1 - c)$. The slight probability of taking a step that gives higher energy is what allows simulated annealing to frequently get out of local minima.

1.4 Travelling Salesman Problem:

The travelling salesman problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? As it is an NP-hard problem simple brute force do not work when problem space is large. Generally it requires a lot of time to find an optimum solution for this problem and given today's computation abilities we generally settle for the solution which is nearer to the global optimum solution as long as the time taken to compute such solution is within acceptable range and that is why it is used as a benchmark for many optimization methods.

2. Project Description:

2.1 Simulated Annealing (SA) for Travelling Salesman problem

We took 20 cities for the simulation purposes so our state space is set of unique the permutation of those 20 cities and these permutation is called tour. The size of our state space is 20! (Finite).

The algorithm starts with a tour which is just a random sequence of 20 cities and will improve gradually along the time. Our algorithm's next state which will be compared and evaluated with current state is obtained by randomly switching two cities in the current tour.

We took total distance that needs to be travelled for certain tour to be energy of the state and our goal is to minimize the energy of the state.

Temperature is the key to SA's ability to find global optimums, and what differentiates it from simple hill-climbing algorithms, is that at each iteration it has a (decreasing) probability of moving to more energy having states. It always moves to more less energy having states but, when next state is more energy having state, it will move to next state when $e^{(energy-newenergy)/T}$ is grater then random (0,1) variable, where T is temperature. Introduction of random variable randomizes the next state which is the key differentiator of simulated annealing from simple annealing.

Theoretical analysis of SA indicates a cooling rate will determine the probability of escaping from local optimums and eventually finding the global optimum because as time passes based on cooling rate with the time temperature decreases reducing the probability of moving to the more energy having states. In our simulation the temperature is $\mathbf{10^4}$ and cooling rate is **0.01%.**

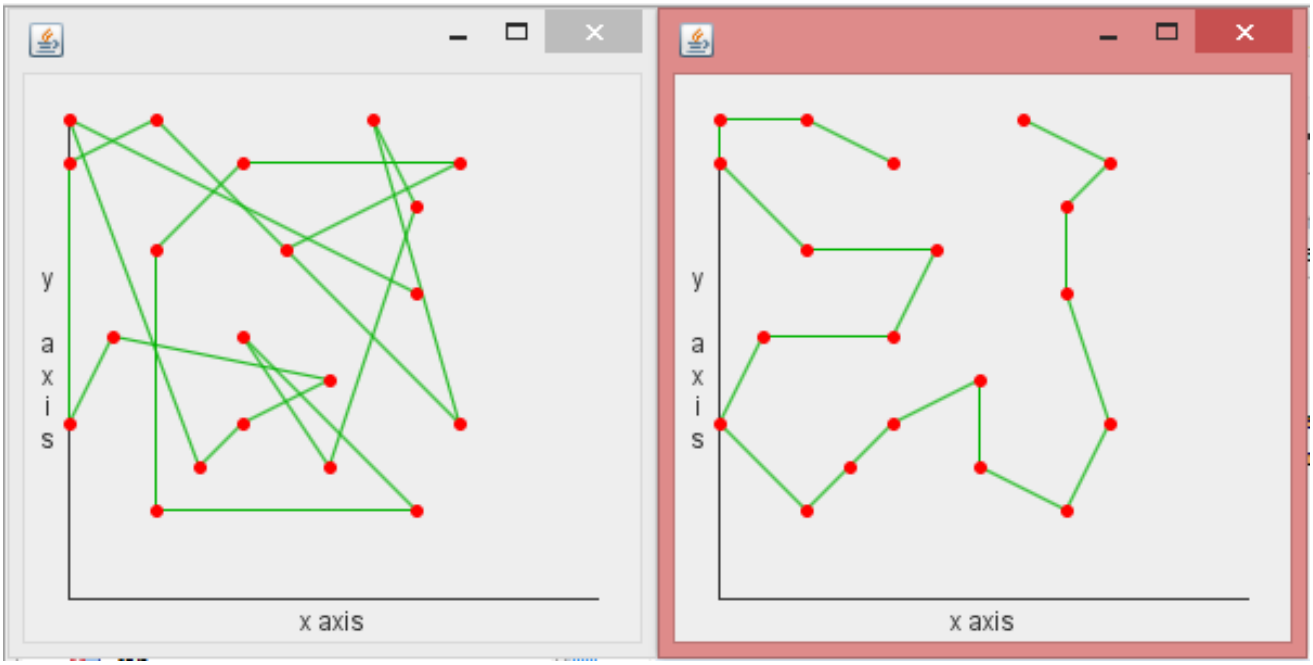
We have divided our project following major sections.

- Generating and managing cities
- Tour management
- Simulated Annealing
- Plotting the results

3. Results:

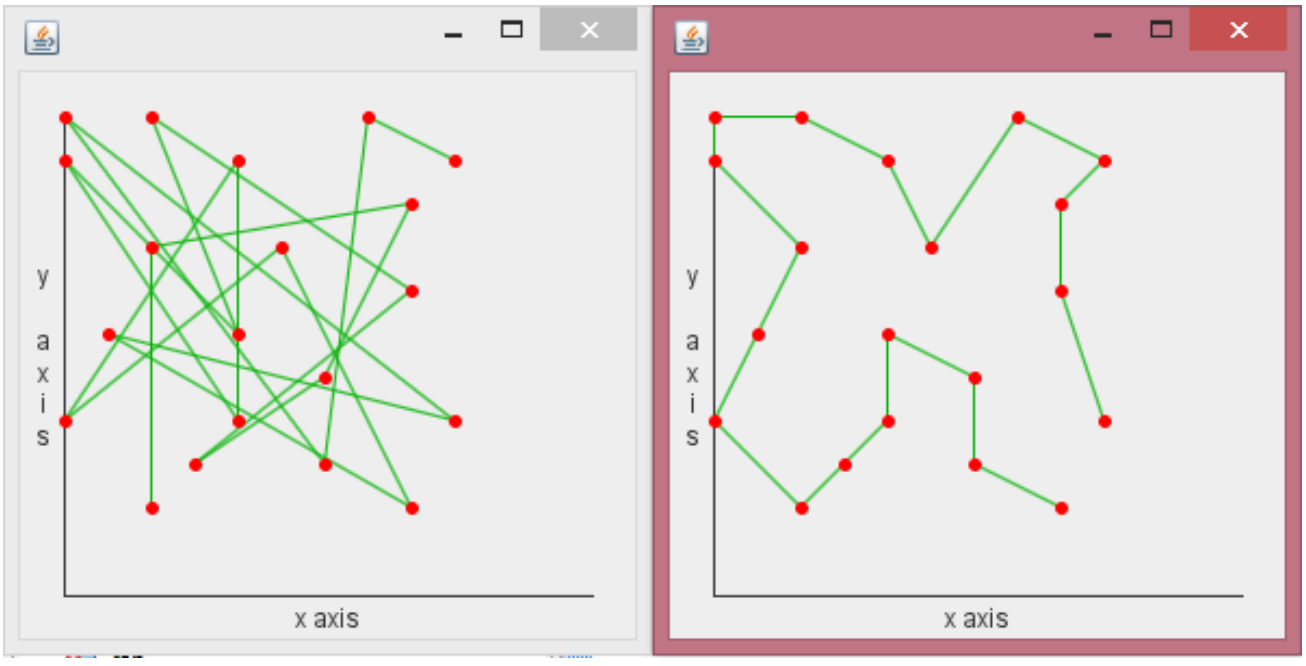
Here, we have displayed results of two different simulations. The best possible solution is a tour having a distance equal to 863 which we got by running the simulation for more than 100 times.

Figure has initial tour and the best possible tour that we got after first simulation.



Initial tour: 2452 Final Tour: 909

This tour shows same result for another simulation



Initial: 2371 Final: 863

Here is the data of sample 10 simulation that we ran.

Simulation No	Initial Distance of the Tour	Best tour distance
1	2616	863
2	2467	863
3	1840	926
4	2011	881
5	2346	916
6	2064	863
7	2579	903
8	2583	947
9	2371	987
10	2452	909

4. Work Distribution:

1. Nikunj Amipara (201001199) – Research on Simulated Annealing, Coding the Simulated Annealing part of code.
2. Parth Shah (201001200) – Research on Markov Chains, Coding City and Tour part of code
3. Om Thakkar (201001203) – Research on Monte Carlo and applications of MCMC, Coding graphical part of code

5. References:

- Markov Chain Monte Carlo Simulation Made Simple by Alastair Smith
- Introduction to Markov Chain Monte Carlo by Charles J. Geyer
- Markov Chain Monte Carlo Method and Its Application by Stephen P. Brooks