Tutorial on Simulated annealing

TSP solve using Simulated annealing

Nidhi Rajkotia Bipinbhai
MTECH CSE (2020-21)
IIIT Vadodara
Gandhinagar, India
202061007@iiitvadodara.ac.in

Dhameja Prashant
MTECH CSE (2020-21)
IIIT Vadodara
Gandhinagar, India
202061003@iiitvadodara.ac.in

Rathod Dharmraj Manojbhai
MTECH CSE (2020-21)
IIIT Vadodara
Gandhinagar, India
202061008@iiitvadodara.ac.in

***Abstract*—This tutorial will be helpful in getting knowledge of Random walk, simulated annealing and how you can solve problem which has huge search space and time complexity is exponentially large. Using simulated annealing, we can solve problems like TSP and K-SAT.**

***Index Terms*—Random walk, annealing, Temperature, energy, probability**

I. Introduction

Simulated annealing is process using which you can solve problem with large search spaces with randomized search technique.we can consider problem like TSP(Travelling Salesman Problem) whose search space is to large it takes exponential time to solve.Using Simulated annealing and non-deterministic search we can solve easily.It does not guarantee optimal solution.

II. Random Walk

In the IHC(Iterative Hill Climbing), we choose series of random starting points followed by deterministic moves. We now look for possibilities for random moves. In which same starting points may lead us to different search space at every different run.
A complete random procedure is called "Random walk". In which search process takes completely random moves disregarding the gradient. The algorithm for that is shown below. So basically random walk explores search space in random

```
RandomWalk()
  node ← random candidate solution or start
  bestNode ← node
  for i ← 1 to n
      do node ← RandomChoose(MoveGen(node,
            if h(node) > h(bestNode)
                then bestNode ← node
  return bestNode
```

Fig. 1. Random walk

fashion. It starts with random node, and initially assigns that node as best node. RandomChosse function generate random successor of that node. And if heuristic value of this node is greater than it takes that node as best node and proceeds further. So, it is completely random.
Random walk relies on exploration of search space. So, the more you explore, the more chances you have of getting better solution or optimum solution.

III. Simulated Annealing

As we all know when we convert gas to liquid and liquid to solid then energy gets down. And material contain minimum energy. So, this process of getting minimum energy state was called Simulated annealing.
We directly don't decrease the temperature, but we cool down slowly so we get a minimum energy state or atoms settle in the minimum energy state. Simulated annealing is an algorithm that combines the two tendencies, explorative and exploitative, of the two search methods. Basic idea is that the algorithm makes a probabilistic random move in any direction. The probability of making move is proportional to gain of that move.
Move is good move, when the probability is high and move is not good move, when the probability is low. Algorithm will make move against the gradient, but probability higher for better move.
Consider maximization problem for state A,B and C, which are given in below figure. Here A and B both are maximas but A is local maxima where B is A global maxima.If algorithm moves to A to C than it has negative gain of ΔE_{AC}.If it goes to B to C then negative gain ΔE_{BC}, which is greater than ΔE_{AC}, so most likely our algorithm goes from A to C. And

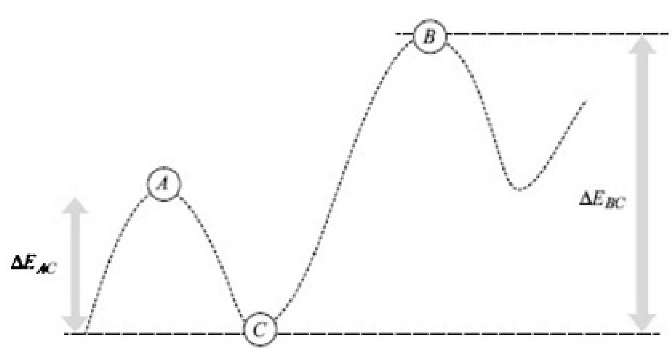

Fig. 2. Maximization Problem

the positive gain from C to B, so most likely to move from C to B rather than A. ”A randomized algorithm that has a simple and constant bias towards better values would be called Stochastic Hill Climbing.”

”Simulated annealing adds another dimension to this”. Generally, It starts with high temperature, it acts like a random walk, and in between it act like gradient, and at low temperature it act like a Hill climbing. But we have to choose temperature parameter widely. It’s not to much high and not to much low. After studied from various sources we found out that for selecting temperature there is no thumb rule. You have to do experiment and find the best suitable value for that.

Now, let’s see simulation annealing algorithm. In below figure algorithm for simulated annealing.

Simulated annealing take probabilistic moves in search space.

```
SimulatedAnnealing()
node ← random candidate solution or start
bestNode ← node
T ← some large value
for time ← 1 to numberOfEpochs
    do while some termination criteria /* M cycles in a simple case */
        do
            neighbour ← RandomNeighbour(node)
            ΔE ← Eval(neighbour) - Eval(node)
            if Random(0, 1) < 1 / (1+e^(-ΔE/T))
                    then node ← neighbour
                if Eval(node) > Eval(bestNode)
                    then bestNode ← node
                T ← CoolingFunction(T, time)
return bestNode
```

Fig. 3. Simulated Annealing Algorithm

Where in local optimization algorithm use heuristic function, here we use Eval function.

Function Coolingfunction decrease temperature after each epoch in which some probabilistic moves are made.One major difference you find out that is it does not do local optimization, like it does not select best neighbour from neighbours.It selects neighbour randomly , then it decides whether it moves towards it or not.

IV. Problem Statement

Travelling Salesman Problem (TSP) is a hard problem, and is simple to state. Given a graph in which the nodes are locations of cities, and edges are labelled with the cost of travelling between cities, find a cycle containing each city exactly once, such that the total cost of the tour is as low as possible.

For the state of Rajasthan, find out at least twenty important tourist locations. Suppose your relatives are about to visit you next week. Use Simulated Annealing to plan a cost effective tour of Rajasthan. It is reasonable to assume that the cost of travelling between two locations is proportional to the distance between them.

An interesting problem domain with TSP instances:
VLSI:http://www.math.uwaterloo.ca/tsp/vlsi/index.html#
XQF131
(Attempt at least five problems from the above list and compare your results.)

V. Problem Solution

As given in problem we assume that our graph is fully connected.We start with initial state as any combination of cities.like let’s say if we have cities A, B, C, D then we can start with any combination(ABCDA,BCADB,...) and we find total cost by calculating euclidean distance from one city to another city and like wise. And we take it as a initial cost.And then in inner for loop we generate random coordinates and we get another combination and then find total cost, if cost is less than we update cost and move forward with this. And if cost is more then we move forward according to probability. For some ephoc it will run and then after come out of this for loop we decrease temperature by some factor.And after that we get some cost to travel.

In Rajsthan problem where our intial cost is 227 and after simulated annealing we got 204.And tour was as given below.
Sujangrah→Sikar→Navalgudh
NeemKathana→Chomu→Jaipur→Dausa
SawaiMadhopur→Tonk→Niwai
Malpura→Kishangarh→Ajmer→Nasirabad
Bijainagar→Bewar→Merta→Makrana
Kuchamancity→Didwana

And i also check for VLSI and i didn’t get same path because it depends on temperature and how you decrease the temperature. So,i got different path and in VLSI it is also different path.

VI. Implementation Repository

The implementation of the Stimulated annealing can be found with the example of Travelling Salesman Problem here.

References

Random walk, Simulated annealing [1].

References

[1] ”A first course in Artificial Intelligence, Deepak Khemani (Chapter 4)”