# Team Brainiacs


| | |
|---|---|
| Vansh Goel | 202051198 |
| Valiveti Swamy Naga Sai Nivas | 202051197 |
| Savan Chaudhari | 202051216 |
| Satyam Gupta | 202051169 |

# Week - 3 Lab Presentation

# Learning Objectives

- **Deterministic Search** - Single path from input to output.
- **Non-Deterministic Search** - Can take many paths, with some arriving at the same outputs, and others arriving at different outputs. (*Used to find approximate solutions*).
  **Example-** execution of concurrent algorithms with race conditions, which can exhibit different outputs on different runs.
- **Randomised Search** - It adds external data to the given input. It may not always produce the correct output. It's used when random inputs have a better chance of producing the correct output (brute force process).
- **Simulated Annealing** - It is multiobjective randomized search (with parameters like temperature, acceptance probability, neighbourhood)

# Problem Statements

- **Travelling Salesman Problem:** 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
- **Rajasthan Tour Planner:** For the state of Rajasthan, find out atleast 20 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.

# Travelling Salesman Problem (Deterministic)

- Given n cities there are n! Combinations possible. Considering the tour to be a complete cycle, and graph to be directed the number of distinct tours becomes (N-1)!/2, which still in the order of n!
- **Brute Force** solution becomes of the order of **n!**
- Another Approach is **Dynamic Approach** of Time Complexity **O(n^2.2^n)**, which is better than n! Of brute approach, but still it is very costly.

# Simulated Annealing

In Metallurgy annealing is the process used to temper or harden metals and glass by heating them to a high temperature and then gradually cooling them, thus allowing the material to reach a low-energy crystalline state.

The same concept is applied to optimization problem, in case of trapped into the local optima, we choose a point outside the minima region, and then we again reach out to minima.

In the Analogy –

- The state space points represent the possible states of the solid;
- The function to be minimized represents the energy of the solid.
- A control parameter c act as temperature.

# Metropolis Algorithm

In the Iterated hill method if the current solution falls in local minima then algorithm remains trapped in the minima, unless you choose a solution outside the minima. That's where Metropolis algorithms comes to rescue:

It states that given two points i, j, if Energy Difference $\delta E = E_i - E_j$ , is positive then State j, becomes new current state, if $\delta E \leq 0$, then the probability of state j becoming current state is given by:

$$P_r(current - state = j) = e^{\frac{E_i - E_j}{k_b T}}$$

T represents the temperature of the solid and $k_B$ is the Boltzmann constant.

Ref. Simulated annealing: From basics to applications,Daniel Delahaye, Supatcha Chaimatanan, Marcel Mongeau

# Simulated Annealing

The acceptance criterion for accepting solution j from the current solution i is given by the following probability:

$$Pr\{\,accept\ j\} = \begin{cases} 1 & if\ f(j) < f(i) \\ e^{\left(\frac{f(i)-f(j)}{c}\right)} & else. \end{cases}$$

# Simulated Annealing

1. **Initialization** $i := i_{start}$, $k := 0$, $c_k = c_0$, $L_k := L_0$**);**
2. **Repeat**
3. **For** $l = 0$ **to** $L_k$ **do**
    - **Generate a solution** $j$ **from the neighborhood** $S_i$ **of the current solution** $i$**;**
    - **If** $f(j) < f(i)$ **then** $i := j$ **(**$j$ **becomes the current solution);**
    - **Else,** $j$ **becomes the current solution with probability** $e^{\left(\frac{f(i)-f(j)}{c_k}\right)}$**;**
4. $k := k + 1$**;**
5. **Compute(**$L_k, c_k$**);**
6. **Until** $c_k \simeq 0$

Ref. Simulated annealing: From basics to applications,Daniel Delahaye, Supatcha Chaimatanan, Marcel Mongeau

TSP Problem

shorturl.at/CGPR8

20 Cities

shorturl.at/jrzKS

DP Solution of TSP

shorturl.at/atwS8

# A modified very fast Simulated Annealing

Authors : Mohammad-Taghi, Vakil-Baghmisheh,  Alireza Navarbaf

Link : https://ieeexplore.ieee.org/document/4651272

# Thanks!

Team Brainiac

Vansh Goel(202051198)
V.S.N.Sai Nivas (202051198)
Savan Chaudhary(202051216)
Satyam Gupta(202051169)