CS594, Python Programming Lab

(https://www.iitg.ac.in/asahu/cs594/)

Assignment V : Finding Optimal TSP using SA

Deadline : 11.55 PM IST, 9th November 2020, **2 marks deduction per day after deadline**

Write a Python Program to find optimal TSP (Travelling Sales Man Problem) tour using Simulated Annealing Meta Heuristics.

TSP Tours: Given a collection of cities and the cost of travel between each pair of them, the **traveling salesman problem**, or **TSP** for short, is to find the cheapest way of visiting all of the cities and returning to your starting point. In the standard version, the travel costs are symmetric in the sense that traveling from city X to city Y costs just as much as traveling from Y to X and in this case it is Euclidian distance between two city X and Y. This problem is know to be difficult problem in ter of complexity and people use heuristics/meta heuristics to solve this problem.

A solution S to problem have all the city in a sequence (in a array with out repetition) and the solution is optimal if the solution have least cost.

A neighbour solution S’ of S have an incremental one swap of two city position in the solution array as compared to S.

Simulated annealing (**SA**) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. SA start with initial random solution S; in each iteration it generate a neighbour solution S’ with one swap of two cities position in the solution, the new solution is accepted if cost of S’ is lower as compared to S, otherwise the solution S’ will be accepted with a probablity (high in lower iterations and low in higher iterations).

Run simulated annealing for 10000 iterations,

Details of SA can be found at : https://en.wikipedia.org/wiki/Simulated_annealing

Data Set: http://www.math.uwaterloo.ca/tsp/vlsi/index.html , xqf131.tsp, xqg237.tsp, ..
The data set contents the points and their location. Distance between two points is Euclidian distance.

Output: final achieved cost and Solution [sequence of cities/numbers)

Submission procedure:

- Create a folder with your name/roll number, put all the source codes and readme files in that folder
- Send your assignments code in compressed folder (tgx/zip/gz) to asahu < at > iitg < dot > ac <dot > in with "CS594: Assignment<V> , < RollNo > " as subject before the deadline
- Please embed comments, how to run and required inputs properly in the code, or a separate readme file.
- Please do not send the provided input files for the assignment
- Submitted code will be checked for Plagiarism