Assignment 6

Due: 30th March.

1 Travelling salesman problem

Finding an optimal solution for certain optimisation problems can be an incredibly difficult task, often practically impossible. This is because when a problem gets sufficiently large we need to search through an enormous number of possible solutions to find the optimal one. Even with modern computing power there are still often too many possible solutions to consider. In this case because we can't realistically expect to find the optimal one within a sensible length of time, we have to settle for something that's close enough.

An example optimisation problem which usually has a large number of possible solutions would be the traveling salesman problem (TSP). In order to find a solution to a problem such as the TSP we need to use an algorithm that is able to find a good enough solution in a *reasonable* amount of time.

- (*50 Points*) Write a code in C++ to solve the TSP, i.e., "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?" by implementing a Simulated Annealing strategy[1]. Also write a python[2] code that call your C++ code, shows shows the initial configuration and then the final path.

Note: You may want to watch this nice visualisations to get an idea of the process:
https://www.youtube.com/watch?v=SC5CX8drAtU

[1] You may want to translate this code https://ericphanson.com/blog/2016/the-traveling-salesman-and-10-lines-of-python/ to C

[2] Use the library subprocess to do it. Something like subprocess.call("./run") would work