Name: Utkarsh Kalra,
Group Number: BS18-DS02
Course: Statistical Techniques for Data Science and Robotics (STDSR) - Spring 2022,
Assignment 3 Code

Contents

1 Structure of the code

The whole code is wrapped in one python notebook with each cell dedicated to a different function.

- The First couple of cells read the data from the **csv** file and sorting the data based on the population and getting the top 30 cities. All of this is being done using the pandas library in python.
- the next cell gives the definition of the class **City** which is designed to carry the information of each city, namely the address, the latitute, longitutde coordinates. So each city will have these 3 parameters. and Then we make a list of cities with each element being an obejct of the class City.
- Function $get_distance_between_two_cities$ returns the distance between 2 sets of coordinates given to it.
- Function get_path_length returns the length of the path given the current order of cities in a list.
- Function $get_proposal_state$ returns a proposal state for the annealing process. It does so by selecting 2 random cities in the given order of cities and reverses the order of these 2 cities.

- Function get_next_state return the next state for the annealing process. It does so by the following criteria.

  If: the path length of the proposed state is less than the path length of the current state then it returns proposed state as the next state.
  Else If: $\exp(Currentpathlength - Proposedpathlength)/T > Random(0,1)$.
  Then the proposed state is returned as the next state.
  Else: return current state as the next state

- Function $do_annealing$ does the whole annealing process by calling the above explained functions in order. It takes in an initial temprature, a cooling rate and an initital state. It continues the annealing process till the Temrature is greater than 1, in every iteration the temprature updates to T = T*cooling rate.

- Function $plot_execution_times_lengths$ plots the plots of cooling rate Vs Time and Cooling rate Vs Final Path length

- Next couple of cells are dedicated for making the animation of the path finding process.

2 Travelling Salesman Problem

In this problem, we intend to find the optimal path with the lowest cost for the path, which is the cumulative summation for the distances between the cities in the path. The path is starting from a city and ends by the same city.

The path contains order of cities with their geolocation (longitude and latitude) and the transformation to their x and y Cartesian Coordinates.

We get the data of the Russian cities from this repository and parse the csv and extract the most 30 populated of this list.

Then, we compute the distances between each city and the other cities. We use the function from geopy.distance to compute the distance based on the geolocation and use the distance in Kilometer (KM)

3 Simulated Annealing

This algorithm is one of the simple algorithms that optimizes the functions without any required computation for the gradients of the function. Therefore, it is considered as a gradient-free algorithm. The algorithm is based on initialize a solution, then proposing a new solution and comparing the proposed solution based on a specific cost and then comparing the cost and act greedy based on the cost.

Here, we define the cost as the cumulitive summation of the distances through the path from the starting city till the end of the path that returns to the same city back.

The algorithm is as following is already explained in the strucuture of code section above.

4 Experiments and Results

Test with temp: 100, 500, 1000
Test with cooling: 0.1 (fast), 0.5, 0.9, 0.99 (slow)
three graphs:
- cooling vs distance (cost)
- cooling vs time
- cooling and tmp with distance

4.1 Cooling Rate Vs Distance and Cooling Rate Vs Time

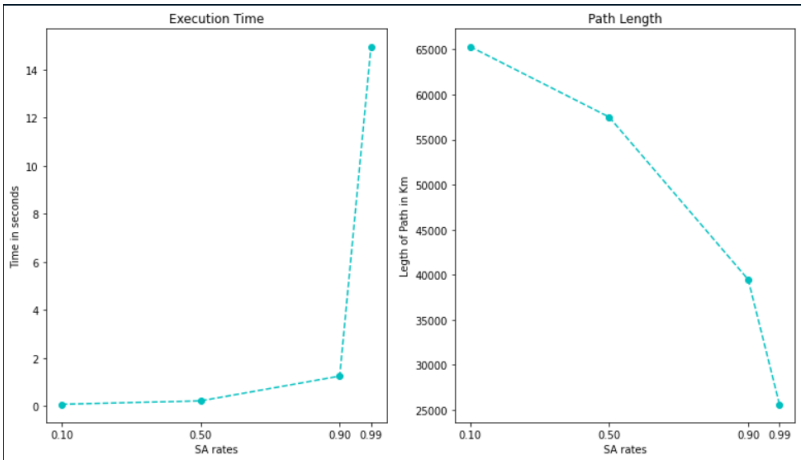

Figure 1: With Initial temprature as 1000

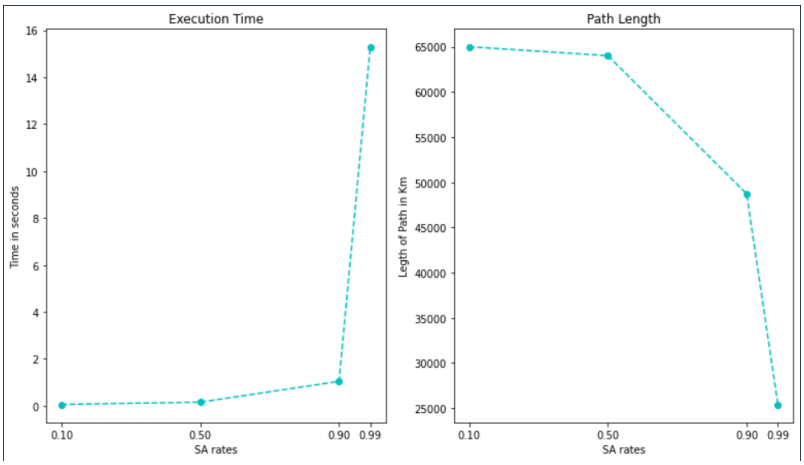

Figure 2: With Initial temprature as 500

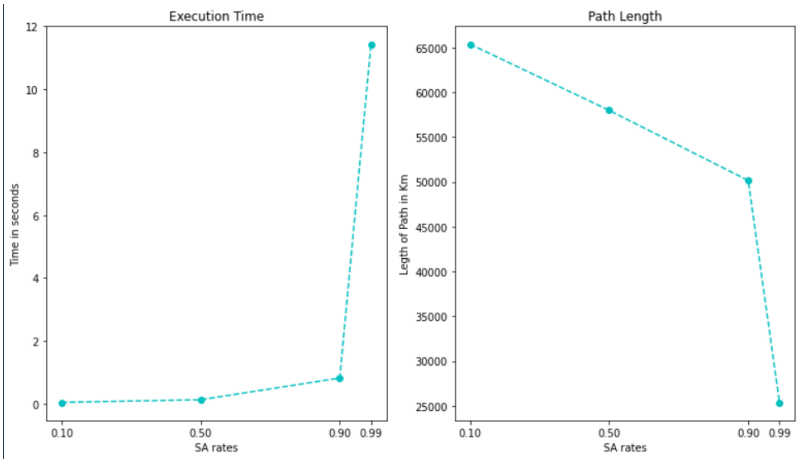

Figure 3: With Initial temprature as 100

4.1.1 Path Length Vs Iteration

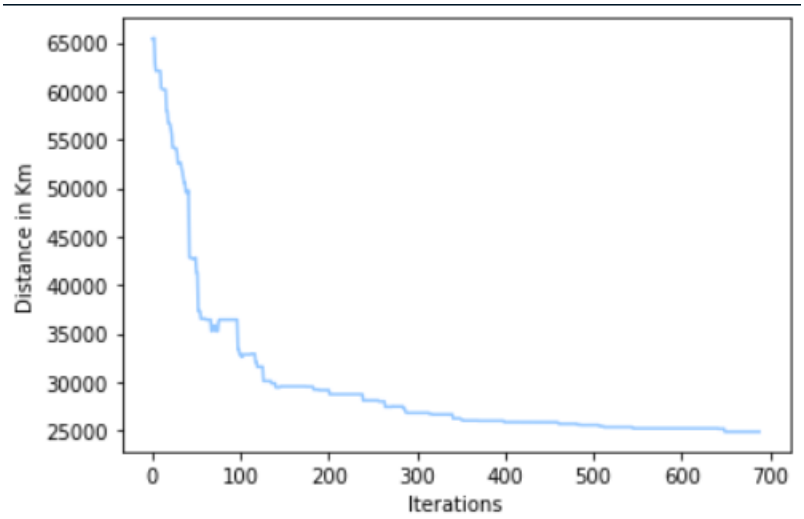

Figure 4: With Initial temprature as 1000 and Cooling rate as 0.99

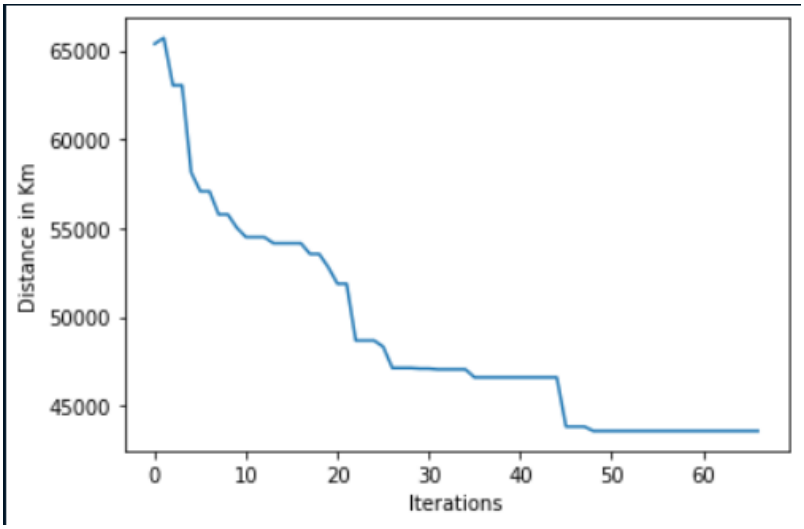

Figure 5: With Initial temprature as 1000 and Cooling rate as 0.9

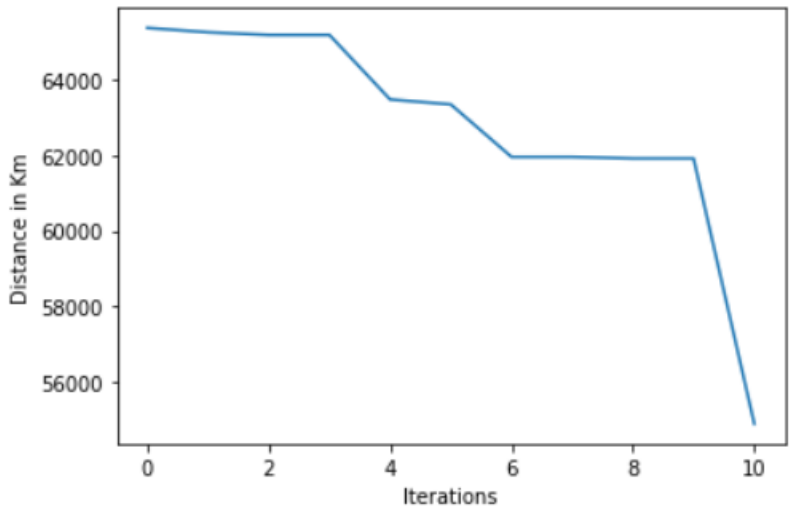

Figure 6: With Initial temprature as 1000 and Cooling rate as 0.5

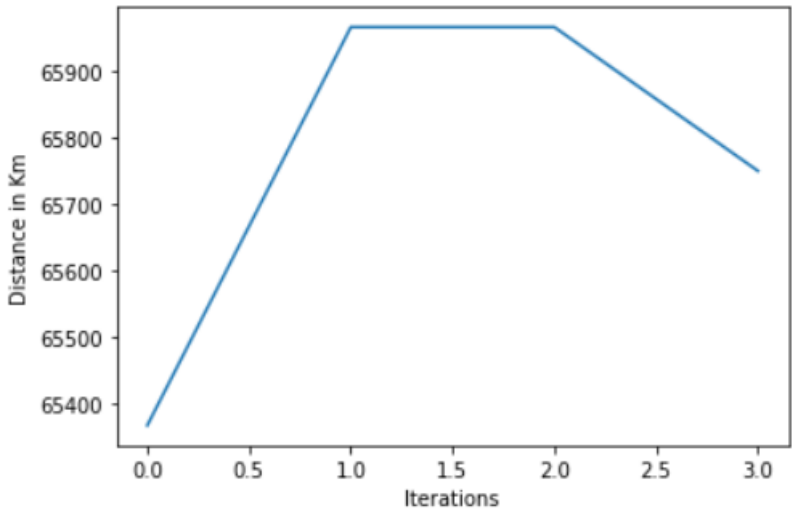

Figure 7: With Initial temprature as 1000 and Cooling rate as 0.1
In this section I have kept the initial temprature as 1000 only as if I decreased the initial temprature the iterations were too less and the graphs were not informational.

4.2 Final selection and result

Finally I decided to take the initial temprature as 1000 and cooling rate as 0.99. So the final graph comes out to be:

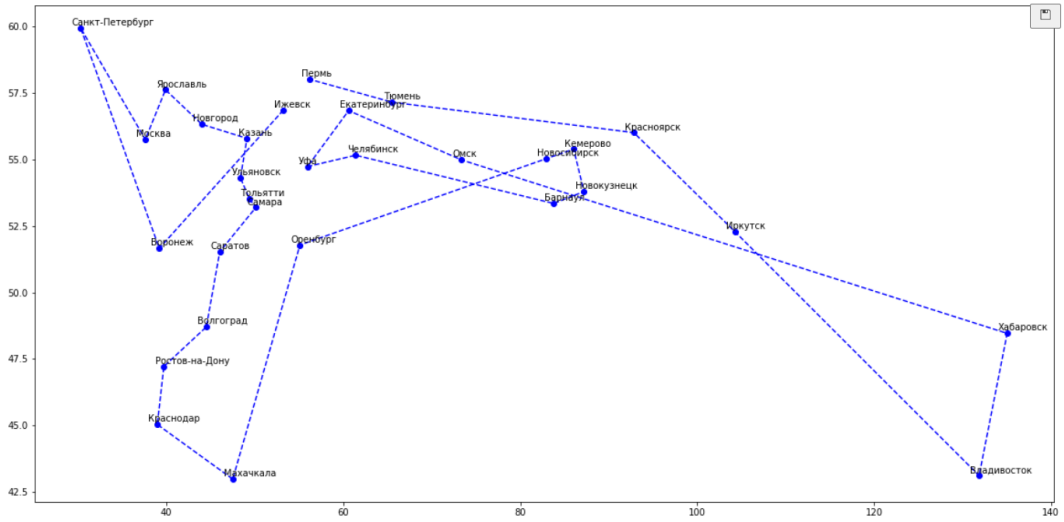

Figure 8: The path length comes out to be 21039.558 Km

4.3 Plotting on the map

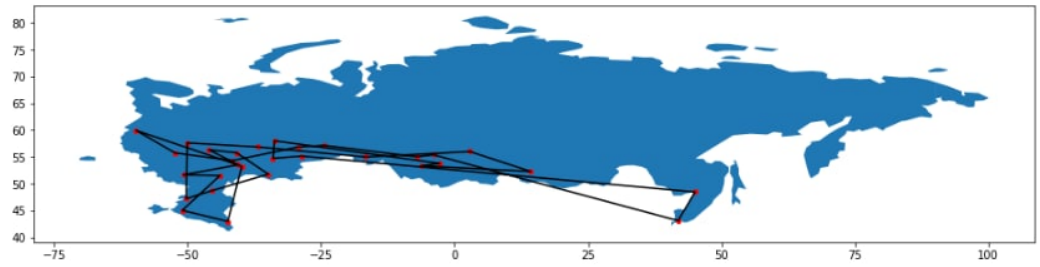

Figure 9: With T = 1000 and cooling rate = 0.83