

Assignment 6

V Manaswini - EE22B065

Import the required libraries such as `numpy` for numerical operations, `matplotlib` for data visualization and `sys` to read the input file from the command line. The python code `travelling_salesman.py` finds the route with shortest possible distance and its length, such that salesman visits all cities exactly once and returns back to the starting point. It makes use of **Simulated Annealing Algorithm** to find the shortest possible route.

How to run the code

1. Place your file(say `cityorder.txt`) containing coordinates of the cities in the directory `ee22b065`.
2. Navigate the terminal to the directory `ee22b065`.
3. Run the command line given below to get the required output.

```
>>> python3 travelling_salesman.py cityorder.txt
```

Description of the code

The code reads the data from a file containing number of cities and their co-ordinates.

“distance(cities,cityorder)” Function

It calculates the total distance traveled in a given city order. It iterates through the list of city orders and makes use of `dist()` function to calculate the distance between consecutive cities. Sum of these distances represents the total distance for the given order.

“tsp(cities)” Function

It seeks an approximate solution using Simulated Annealing Approach, it takes the list of cities as input. The steps involved are:

- It selects any random city as starting point from the given list of cities and proceeds for iteration, and this will be the starting point for every iteration.
- Assuming the value of temperature analog(t) to be 1000, with a decay rate of 0.95 for each iteration.
- It then iteratively explores neighbouring solutions by swapping two cities in the route.
- If a neighbouring solution has a shorter distance, or if it's accepted probabilistically(as $e^{-\frac{\Delta Distance}{t}}$) based on the current temperature analog, then the path will get updated.

- The temperature analog is reduced in each iteration, and the process continues until the temperature reaches a minimum threshold of 10^{-15} .
- Finally, it returns the best possible path found so far.

Visualization

- The code utilizes `matplotlib` to visualize two paths, random path and the shortest path found by simulated annealing algorithm.
- Running the code produces two plots(Random Path and Shortest Path).

Percentage Improvement

- First, it generates a random route and calculates the length(R) of this route.
- Then using simulated annealing algorithm, it calculates the length(S) of the shortest possible route found so far. Finally, output the percentage improvement $\left(1 - \frac{S}{R}\right) \times 100$.

Observations from the Given Input File

Given file `tsp40.txt` contains 40 cities with their coordinates. The best shortest possible total distance after 10 repetitions is found to be 7.176061992241605, and the City Order for the Shortest Path Followed is [5, 29, 0, 9, 39, 22, 10, 38, 34, 3, 17, 19, 11, 14, 15, 23, 25, 16, 1, 36, 6, 31, 26, 18, 24, 33, 2, 7, 32, 35, 8, 27, 13, 30, 20, 21, 4, 12, 37, 28]

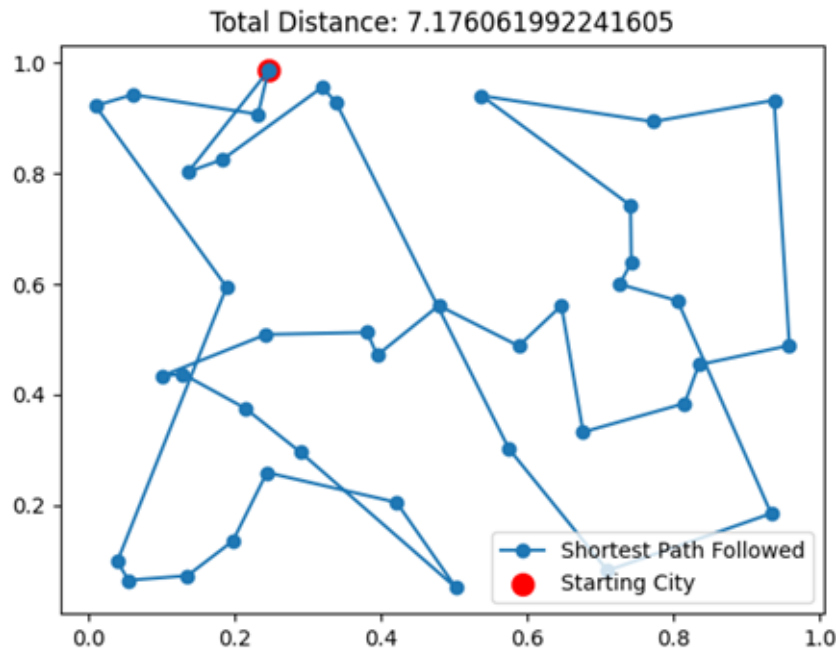


Figure 1: Shortest Route