ASSIGNMENT 02. TRAVELING SALESMAN PROBLEM

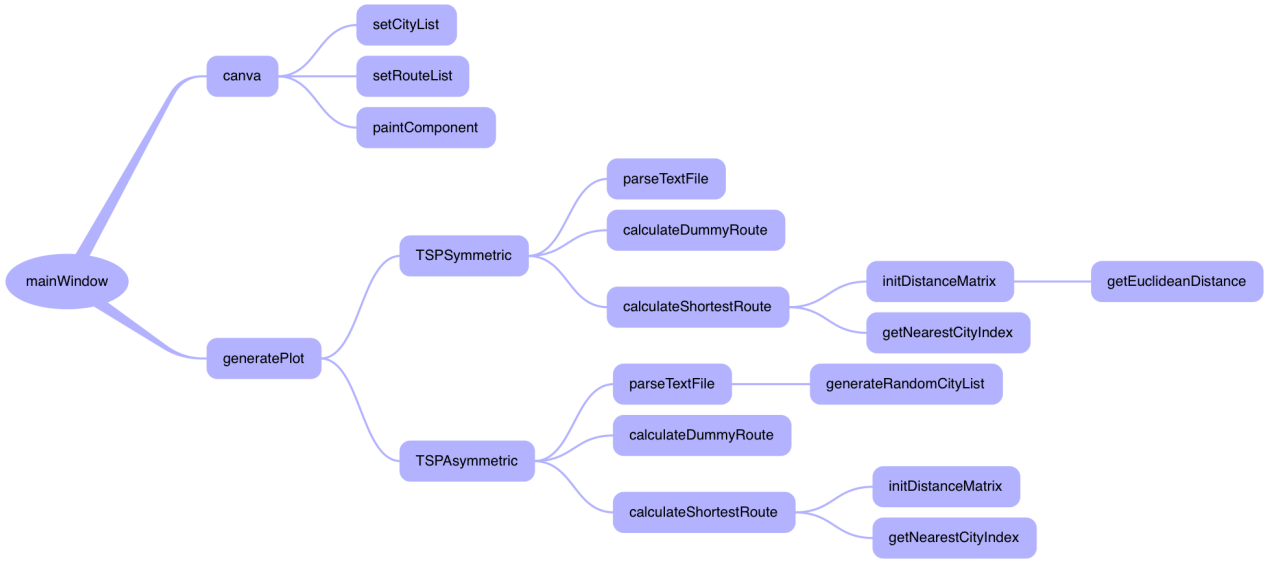
1. INTRODUCTION

This program implements a GUI to display the travel route between a group of cities. It supports the input of both symmetric and asymmetric data and allows selection between a dummy route (by connecting the cities in order) and the shortest route (traveling salesman greedy algorithm). It also displays the total circuit distance for the input based on the selected route type.

2. INFORMATION HIDING

- The `TSP` class, along with its derived classes `TSPSymmetric` and `TSPAsymmetric` hide the parsing of file data as well as the algorithm behind generation of the dummy and shortest route through functions `parseTextFile`, `calculateDummyRoute` and `calculateShortestRoute`.
- The `Canva` class abstracts the drawing functions. It exposes setters for city list (`cityList`) and route list (`routeList`). When the route list is set, a repaint of the drawable area is automatically triggered to reflect the changes.
- The `Route` and `City` class declare the variables as private and provide public getter and setter methods to view and modify the values of the variables.

3. STEPWISE REFINEMENT



4. MODULES

The project consists of three main modules:

1. `MainWindow` – The GUI i.e., the window frame along with its components such as `Canva`, Buttons, etc. with responsibility for laying out the components onto the window frame and defining the interactions and behaviours for the components.
2. `Canva` – The drawing area with responsibility for plotting the data (cities as points and routes as lines).
3. `TSP` (along with `TSPSymmetric` and `TSPASymmetric`) – This module has the responsibility for reading and parsing the data and then evaluating the dummy and / or shortest route.

The `MainWindow` module calls the `Canva` and `TSP` modules. It first calls the `TSP` module to parse the file data and then evaluate the route. After the evaluation is complete, it passes on the route information generated by `TSP` to `Canva` for plotting.

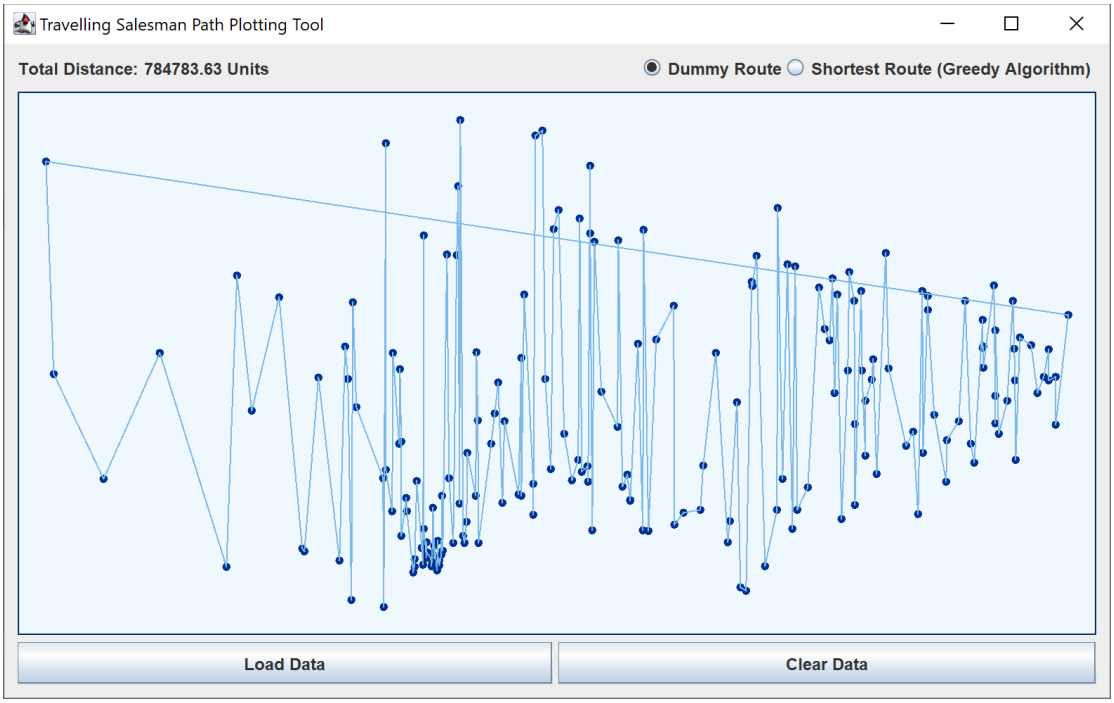
5. SAMPLE OUTPUT

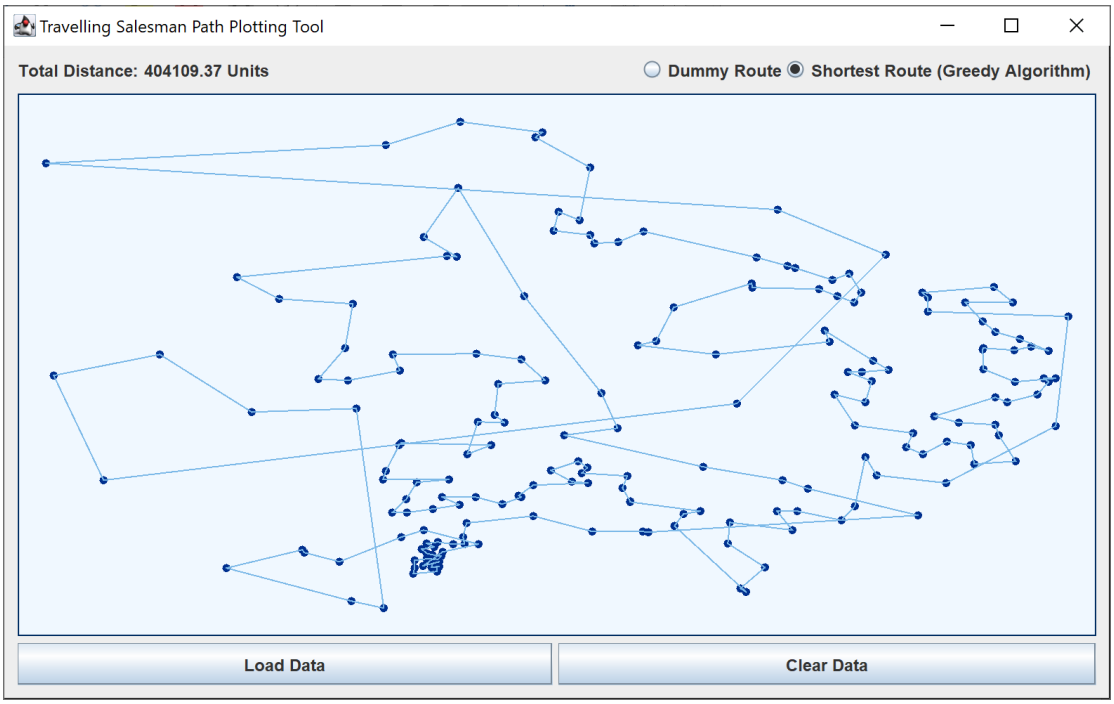
This section displays the output for dummy route and shortest route (using Travelling Salesman Greedy Algorithm) for both symmetric and asymmetric data. For each category of data, three sub-categories of output are presented based on the size of the data set i.e., Small, Medium and Large.

The source for symmetric data is: https://www.math.uwaterloo.ca/tsp/world/countries.html
The source for asymmetric data is: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp/index.html

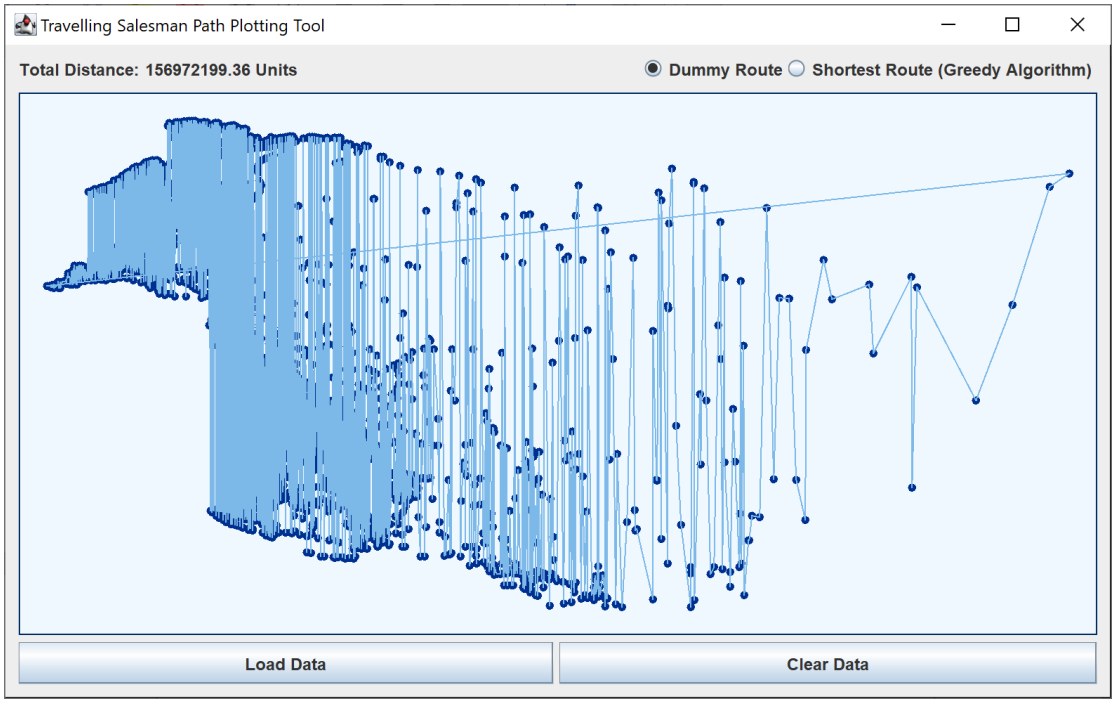
5.1. Symmetric Data:

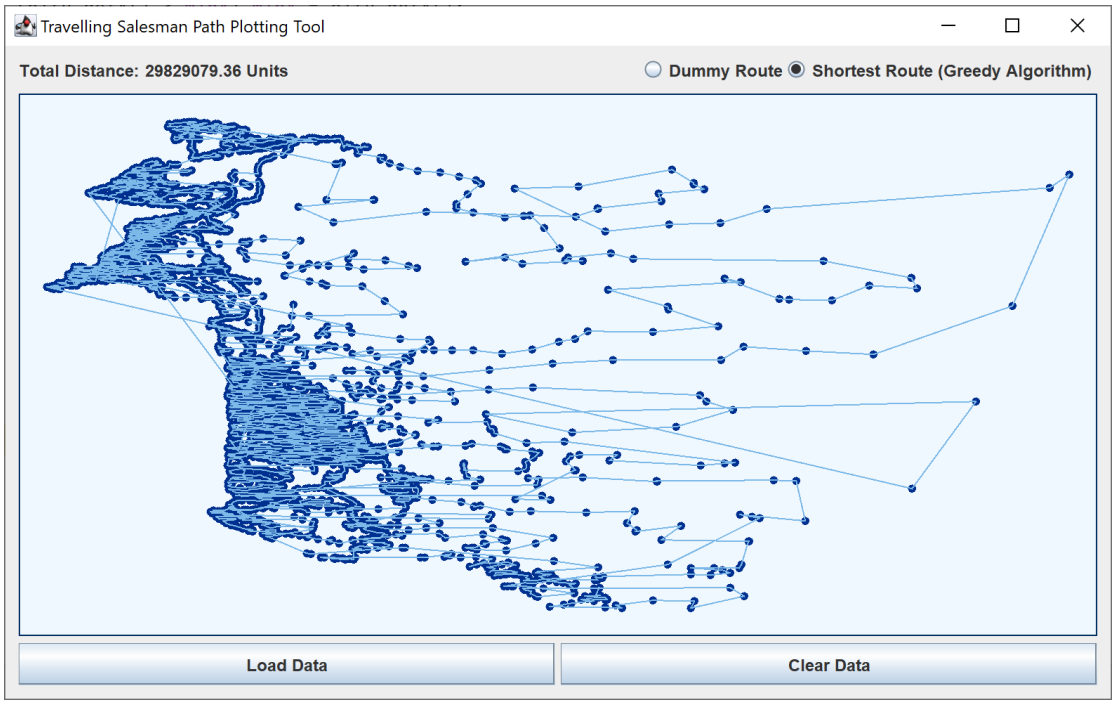
5.1.1. Small Data Set (Qatar) – 194 Cities



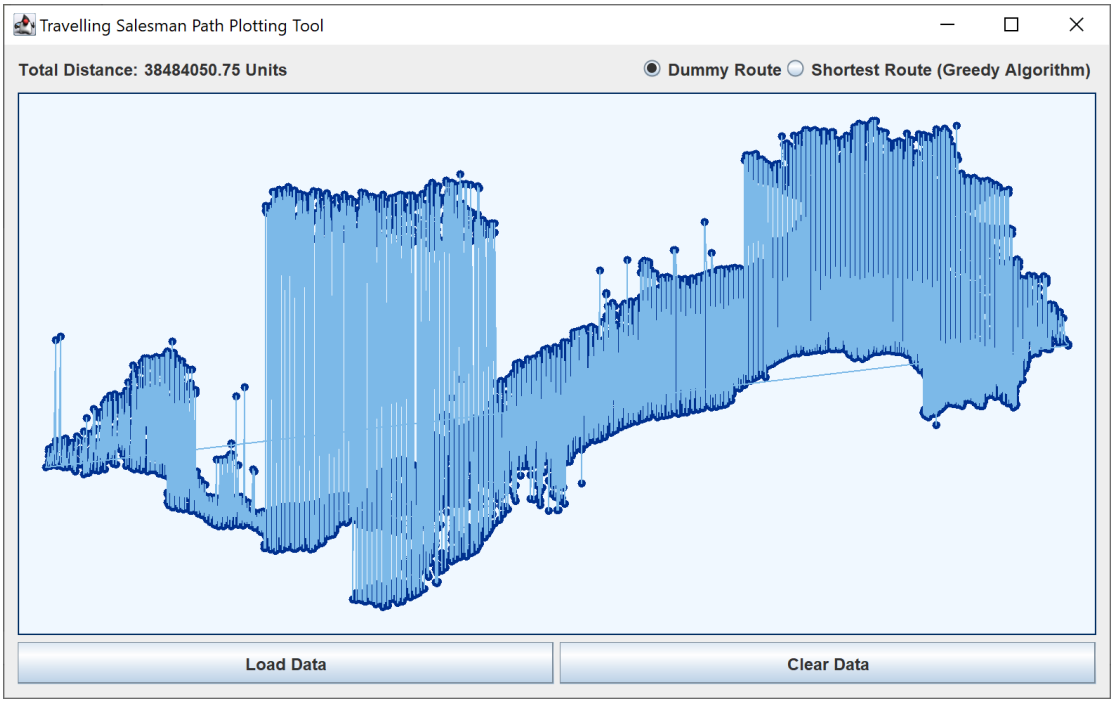


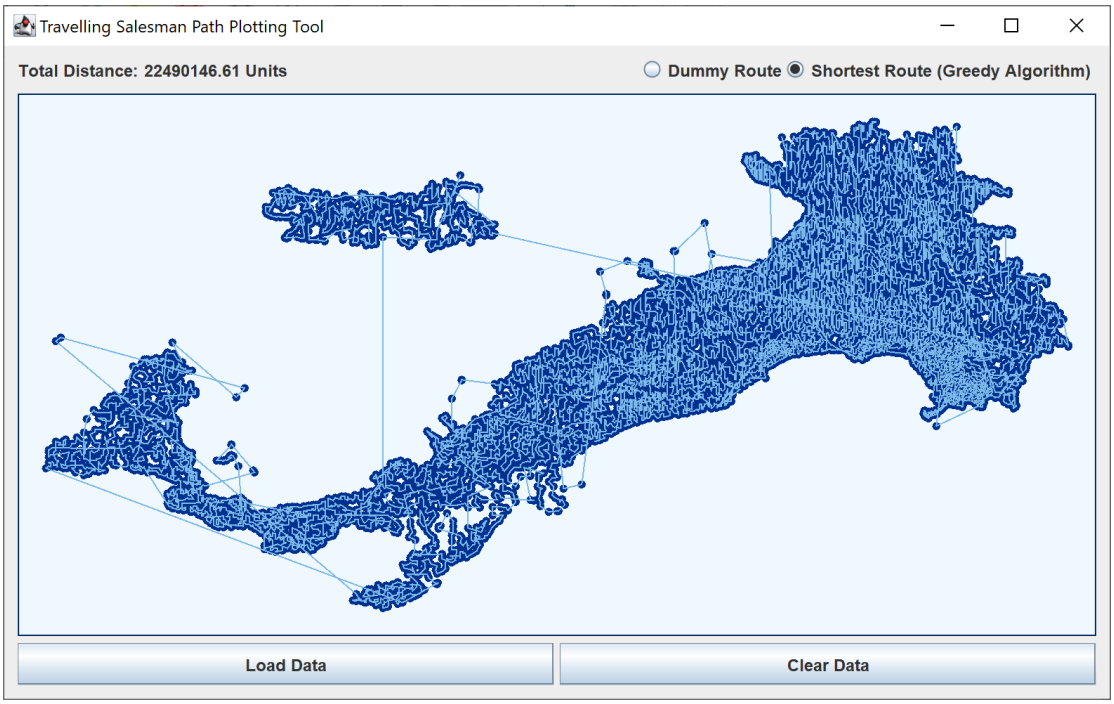
5.1.2. Medium Data Set (Canada) – 4,663 Cities





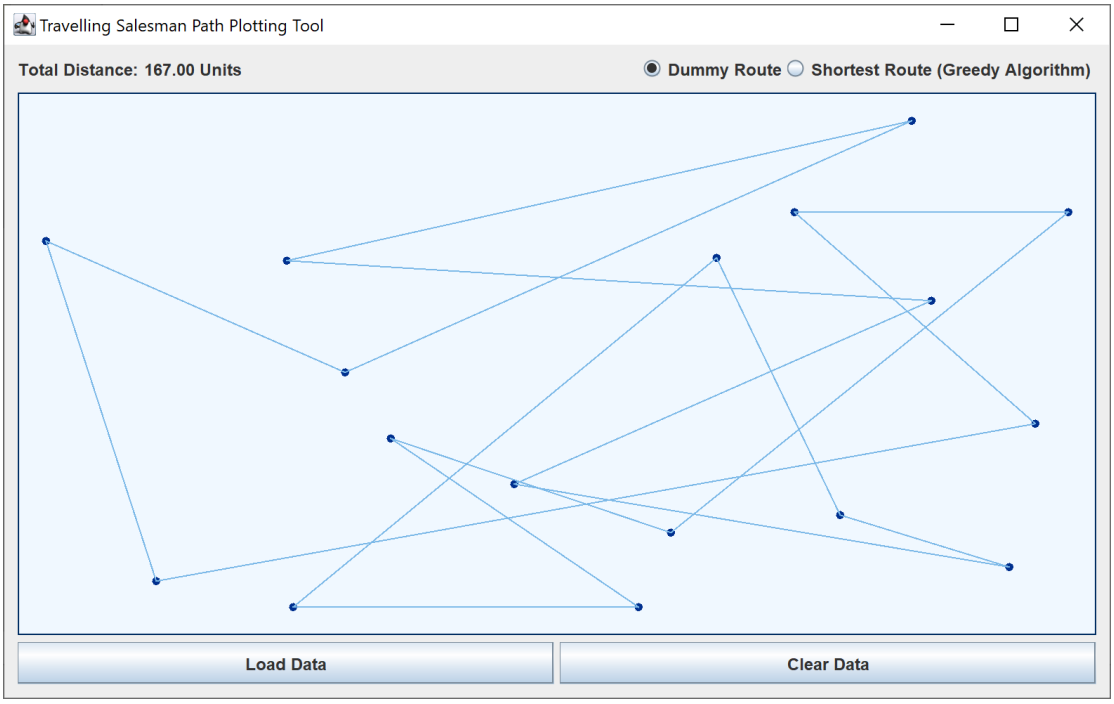
5.1.3. Large Data Set (Italy) – 16,682 Cities

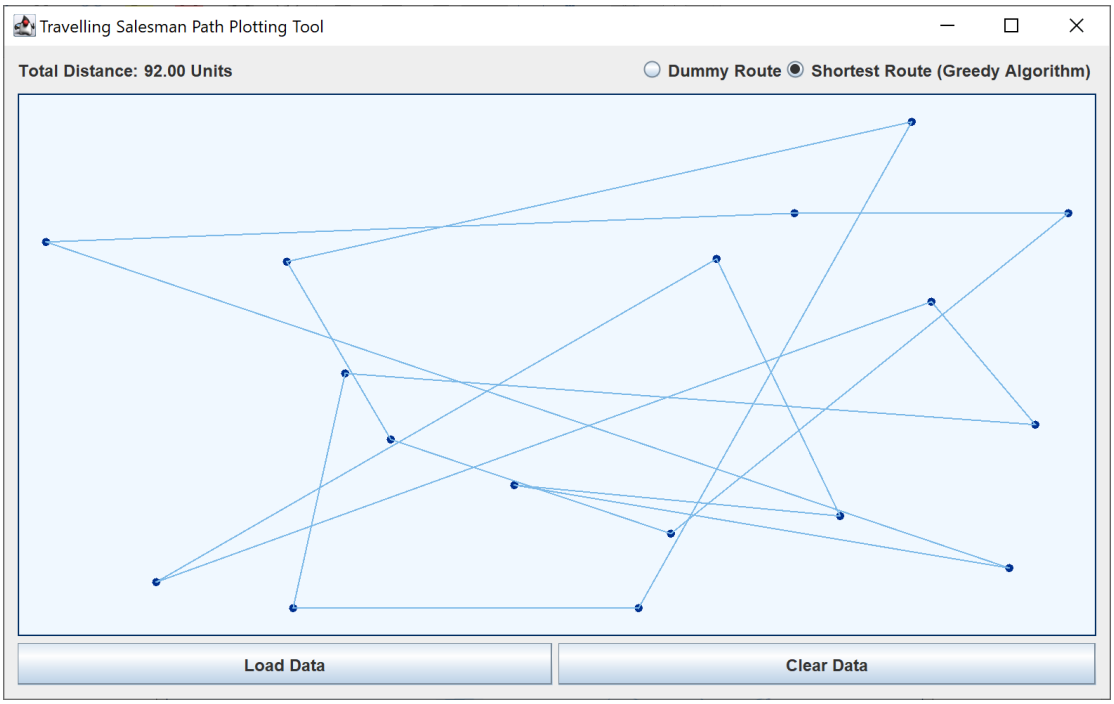




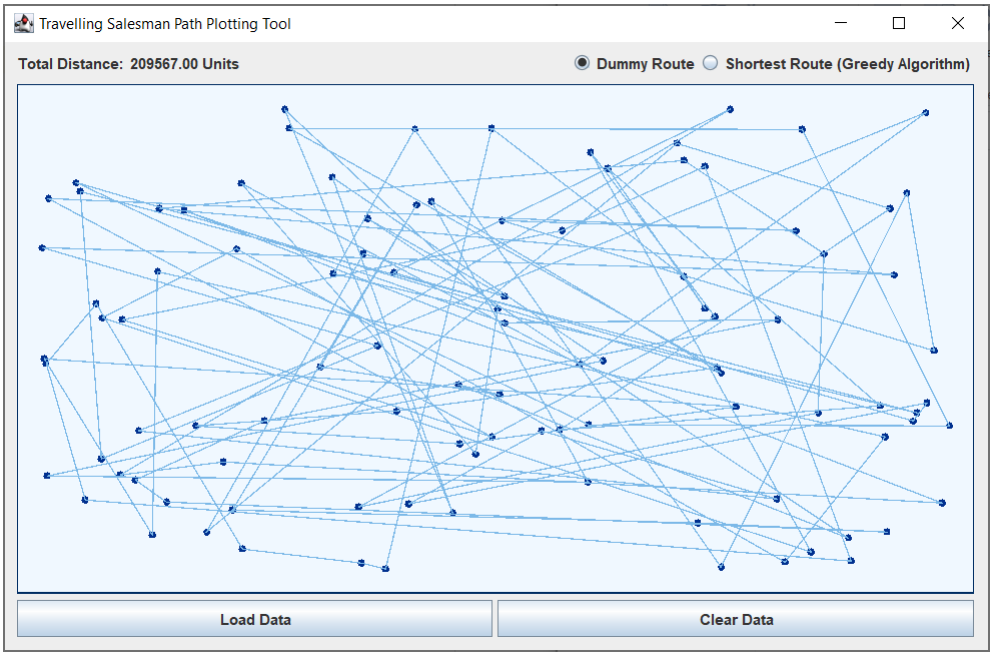
5.2. Asymmetric Data:

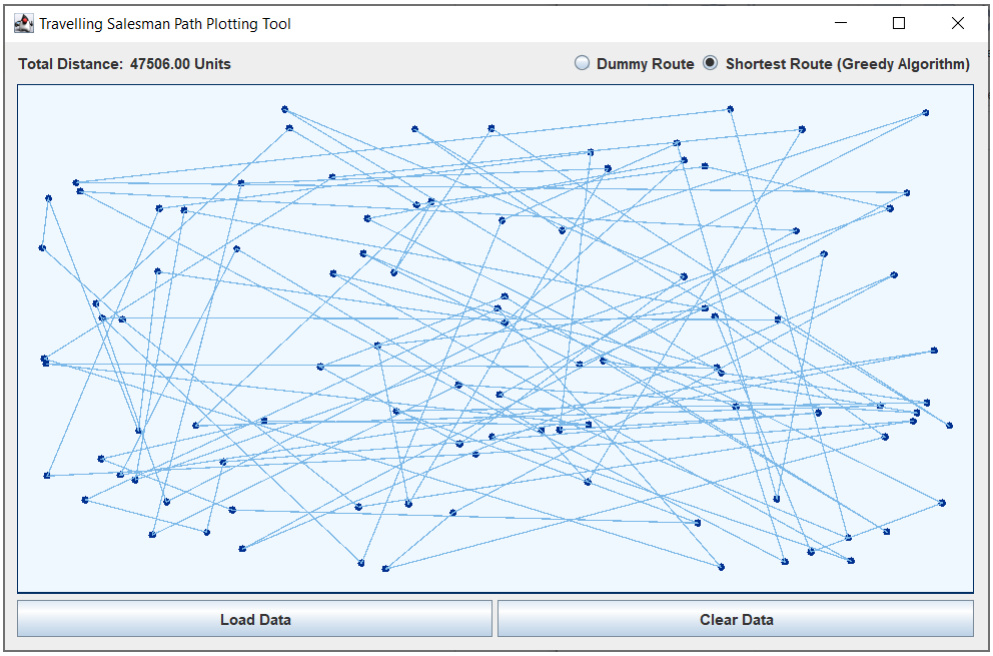
5.2.1. Small Data Set (br17) – 17 Data Points



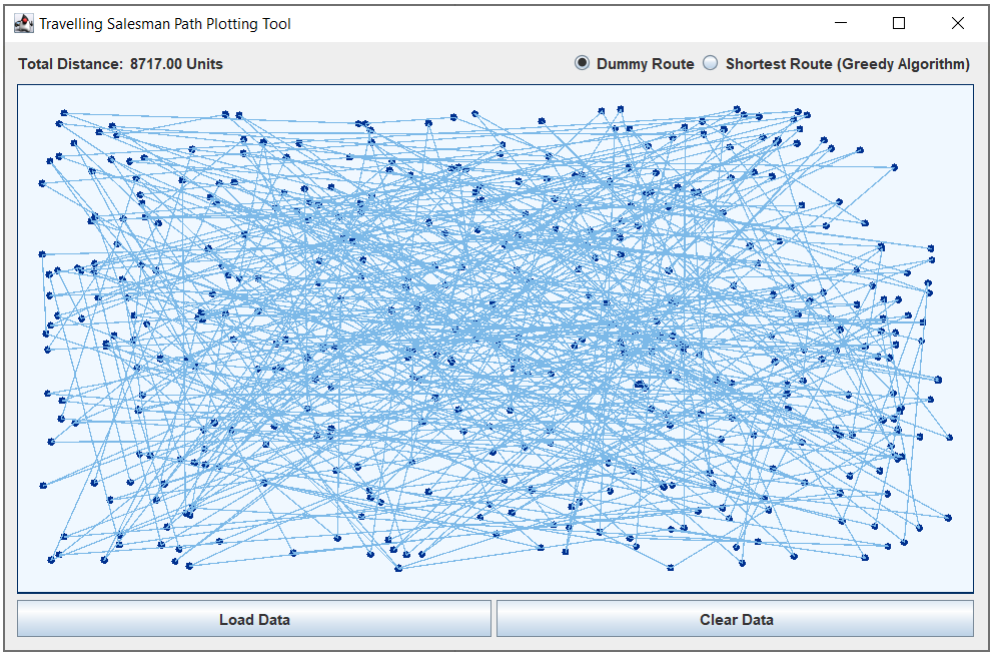


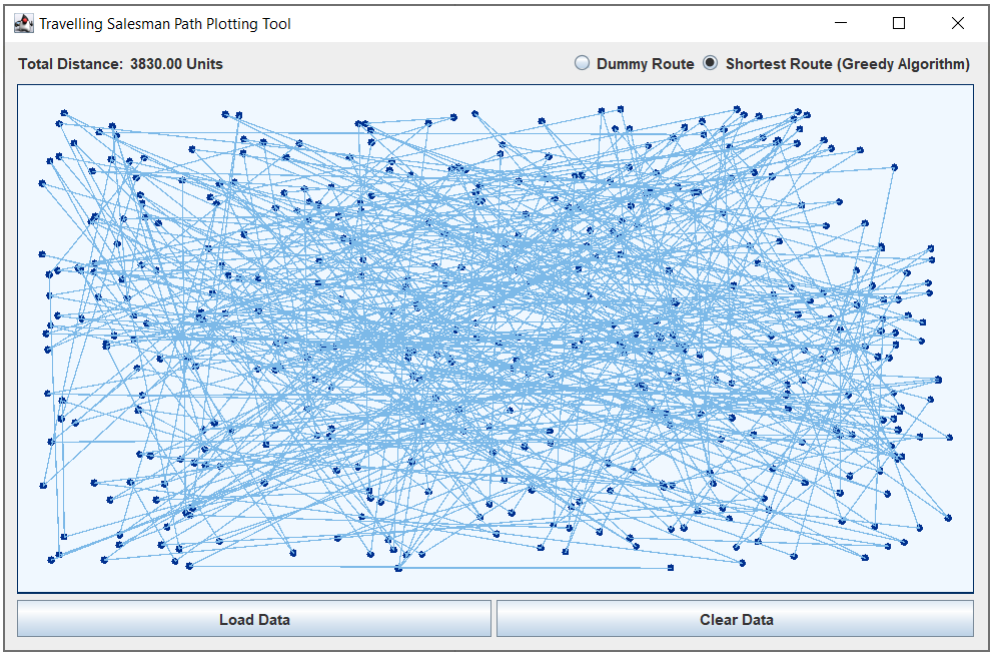
5.2.2. Medium Data Set (kro124p) – 124 Data Points





5.2.3. Large Data Set (rgb443) – 443 Data Points





6. LIMITATIONS

- For symmetric data, Italy with 16,682 cities is the largest file (from the given data repository for symmetric data) that our program can process to calculate shortest route. This is because of the requirement for a large amount of heap memory to store the distance matrix used in the Travelling Salesman Greedy Algorithm. However, we can get the dummy route for even the largest data set i.e., China with 71,009 cites as it does not require computation of the distance matrix.