

Application for Facility Location Problem in Waste Management
User documentation

Albert Khaidarov

UP FAMNIT

August 2021

1. Introduction

The program was fully developed in Java. MPJ library was taken to get MPI functionality and JavaFX was used as GUI library. As for GUI, apart from JavaFX, WebView was used, as well as JavaScript technologies to display the map.

1.1. Purpose of the application

The aim of the application is to find a suboptimal placement of facilities set that processes the accumulated waste, collected from garbage collection spots. The appliance of the algorithm is supposed to optimize the distance between accumulation sites and processing plants.

1.2. Requirements

It is recommended to use the IDE (integrated development environment) to run the program for the Java programming language, e.g. IntelliJ IDEA, Eclipse, etc. As GUI part of application is written in Node JS programming language need to be installed node and npm packages for JavaScript.

2. Instructions and use case

Before running the program, a user should start a server from localhost. To do this we need to launch the commander and start a server by typing the command “*npm start*”:

```
C:\Users\User\newproject>npm start
> start
> parcel index.html

Server running at http://localhost:1234
√ Built in 12.21s.
```

Figure 1: Starting the Node JS server

After we can launch IDE and start the Java program in the Main class:

```
public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("osm_map_main.fxml"));
        primaryStage.setTitle("OSM Assignment Sites - K-Means");
        primaryStage.setScene(new Scene(root, w: 800, h: 600));
        primaryStage.show();
    }

    public static void main(String[] args) { Launch(args); }
}
```

Figure 2: Main class in Java

Thus, a user launches a java program. A window opens, where we can choose an area by zooming, scrolling, moving. Here is the same principle as in web-applications with GPS-coordinates like Google Maps, OpenStreetMap, etc.:

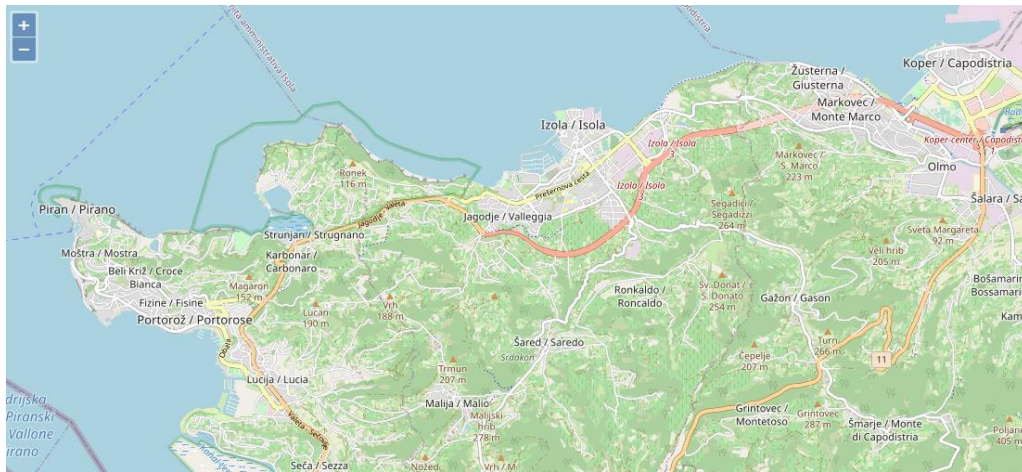


Figure 3: Displaying the map in WebView

2.1. Setting parameters

Before choosing appropriate area user should set three parameters:

- Number of clusters
- Number of points
- Number of iterations

Amount of points will correspond to the number of garbage collection spots, and quantity of clusters – how many processing plants to build. Third parameter representing number of iterations of the algorithm.

All these parameters affect the execution time of the algorithm.

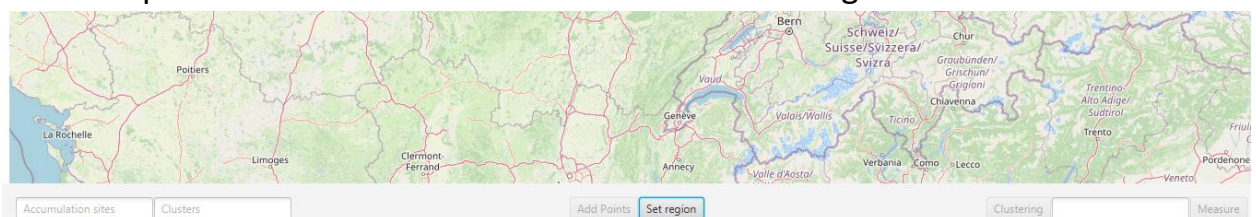


Figure 4: User panel

2.2. Selecting area

After setting all the parameters, user can click the "Set region" button and with the left mouse button we draw a circle to choose needed area on the map:

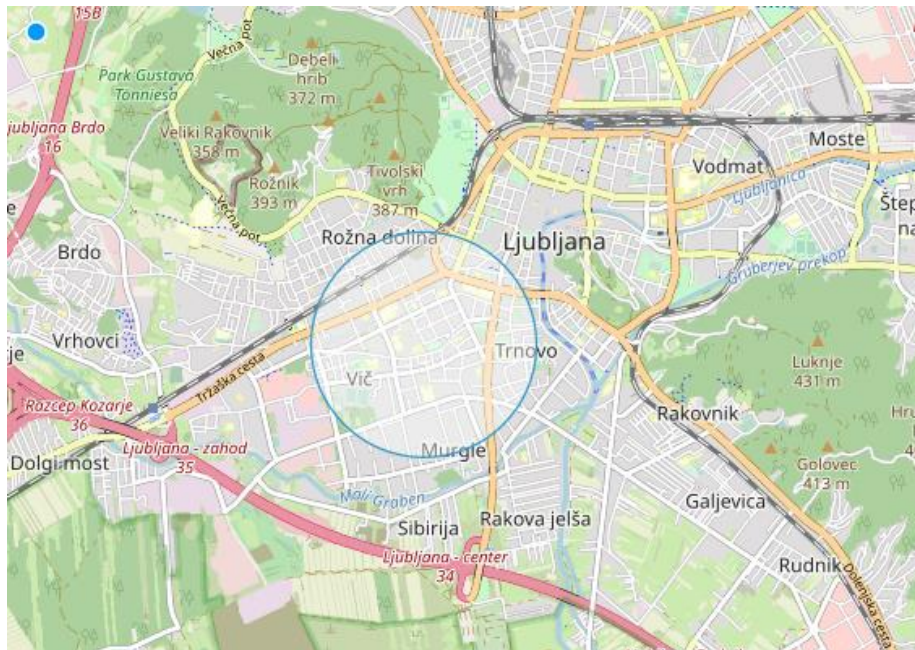


Figure 5: Selecting region for computation

2.3. Clustering

After selecting a region, a user can add garbage collection points to this area by clicking the “Add Points” button. Each displayed point on the map will be randomly assigned a coordinate and capacity (tons of waste per year). For starting the algorithm click the “Clustering” button. Result will be shown on the screen:

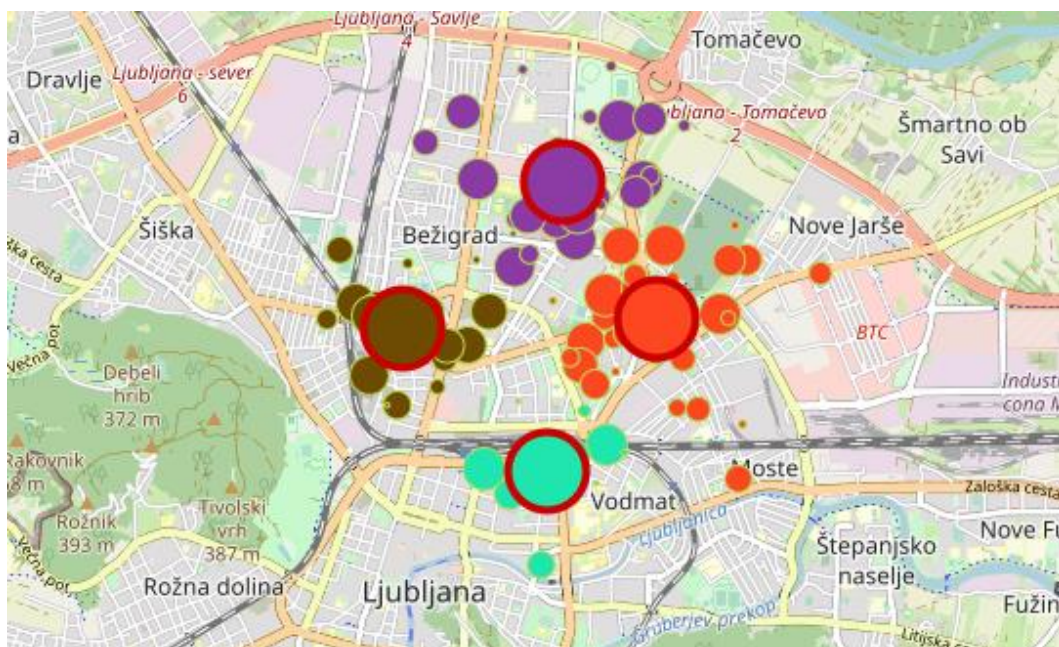


Figure 6: Clustering result

Each set of waste collection sites (clusters) has its own color, and the red-framed circles inside each cluster represent the optimal location of the recycling facilities.

2.4. Measuring

To obtain data on the coordinates and speed of the algorithm, press the "Measure" button. Immediately after that, a CSV file with the data for the test will appear in the project folder.