Department of Electrical and Computer Engineering
University of Tehran

Operation Research

Final Project

Optimal Vehicle Routing

TA: **Niusha Mirhakimi**

December 2021

Contents

1 Abstract **2**

2 Part 1: Dataset Generation **3**
2.1 Data Extraction 3
2.2 Model Graph 4

3 Part 2: Minimum Cost Routing **4**
3.1 Programming Tips 4

4 Notes **5**

1 Abstract

In this project, we aim to combine the knowledge we gained in the course and your Python programming skills to design and solve a real-world optimization problem. The problem is to find the best rout from one point to another in the Tehran with known places. This project consists of two parts. First, we collect the required data (e.g. places' names and coordinates). Second, we define the optimization model using the prepared data in the first part in order to find the best rout between two distinct places with the minimum cost in terms of distance, traffic, etc. The procedures of implementation of Part 1 and Part 2 are described in detail in this manual.

2 Part 1: Dataset Generation

For this goal, we use the Google map of Tehran which features a desirable UI and accurate reports of distances and traffic status of routes.

2.1 Data Extraction

Consider at least ten places, including university campuses, hospitals, subway stations, etc, only in district 3 and district 6 as shown in Figure 1. Please try to choose such places in the map that covers the districts perfectly. Then, extract the following properties for your chosen locations:

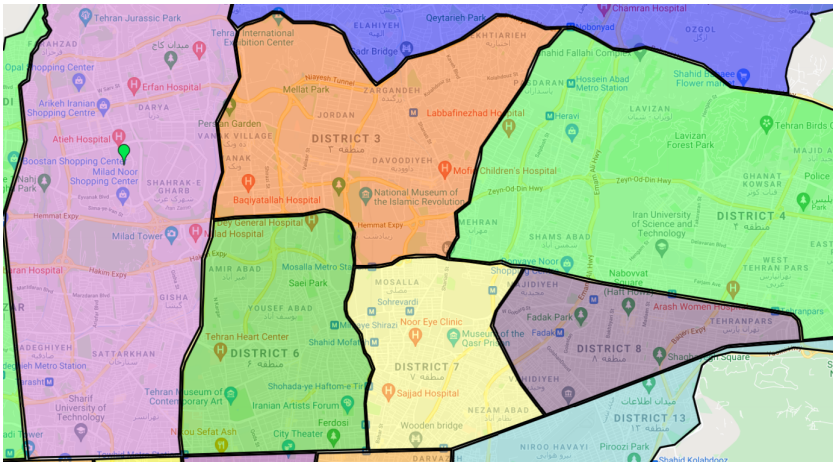

Figure 1: The map of some Tehran districts

- **Place_index**: The index of the place (i.e. assign each place a number)
- **Place_name**: The Places' names in English
- **Latitude**, **Longitude**: The coordinates of the places
- **Neighbors_indice**: Indices of the neighboring places
- **Neighbors_weight**: The weight (cost) between two distinct places which is a number according to the Google map reports of distance, time, and common traffic status.

Afterward, store the above-mentioned properties of all places (nodes) into a *csv* or *json* file. Please note that this file will be read and used to design your optimization model, thus, try to make it clean and precise.

2.2 Model Graph

In this step, you can use either Microsoft Visio software or DrawIO website to draw a graph of your model, including nodes' names, indices, and weight of each rout. As an example, a sample graph with five places is shown in Figure 2.

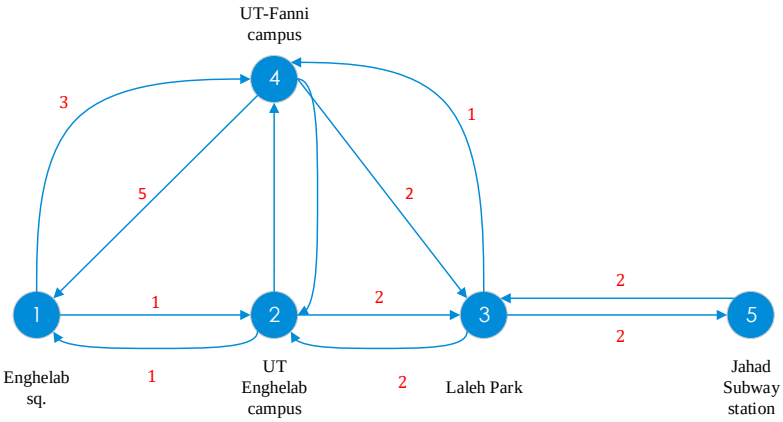

Figure 2: A sample graph of five places in Tehran district 6

Save your implemented graph as an image and represent it in your report along with a picture of the map with the highlighted locations of your chosen places.

3 Part 2: Minimum Cost Routing

In this part, you aim to model your own optimization problem with respect to the costs and constraints. Clearly, explain your model specification and its constraints in the report.

After determining your model and the constraints, write a Python script that calculates the optimal rout between any two distinct places using the prepared data of Part 1. Your Python code should provide a simple console user interface for users which reads the given origin and the destination, and then returns the optimal rout. The outputs of the code are:

- The indices of the origin and the destination along with their coordinates
- The indices of all places on the way of the optimal rout
- The total cost of the optimal rout

3.1 Programming Tips

- ***Note 1:*** The style of your code is an essence of your work. Try to write clean and legible code in Python based on information available on the Internet. For instance, the names of the variables should be meaningful or indentation and spacing should be integrated in the code. Also, consider the encapsulation code for the different sections.
- ***Note 2:*** You can use either *pulp* or *mip* libraries in Python to model and solve your optimization problem.

4 Notes

1. Submission materials:

   - **Python code**: A folder containing the script requested in the project manual. The folder should be named "code". You may also use Jupyter Notebook for the Python code.
   - **Dataset**: A single *csv* or *json* file containing all the required extracted data from Google map in Part 1.
   - **Project report**: A single PDF containing all descriptions, graph, results, and code output screenshots.

2. The composition of final mark:

   - Python code 30%
   - Dataset 30%
   - Project report 40%

3. Cheating, plagiarism, and unauthorized collaboration are strongly prohibited in this course and any non-compliance would be treated in accordance to the academic integrity.

4. If you have any questions regarding this project, please contact me via niushamirhakimi@gmail.com.