Testing location optimal solution sensitivity

Table of Contents

Description

The **weighted p-median** and **weighted p-center** are facility location problems with significant applications in operations research. They are commonly used for placing facilities such as ambulance stations, hospitals, fire stations, or warehouses within a given region.

The main objective of this project is to **analyze the sensitivity** of the optimal solutions of the weighted p-median and weighted p-center problems when certain **input parameters of the transportation network (graph)** are modified.

Goals of the project:

1. **Implement an appropriate algorithm** capable of finding an optimal solution for the weighted p-median or weighted p-center problem.
2. **Program mathematical expressions** modeling changes in the input parameters of the given graph.
3. **Test and evaluate** these models on small and medium-sized graphs.

Download and install

Python interpreter

To run this application, you need to have Python installed on your system. I recommend using the latest stable version of Python for the best compatibility. For this project, I used Python version 3.13.2.

You can download the latest version of Python from the official Python website:
https://www.python.org/downloads/

To verify that Python is installed correctly, you can run the following command in your terminal:

```
python --version
```

If Python is installed, this will return the installed version number.

Cloning the repository

To get started with the project, you'll need to clone the GitHub repository to your local machine.

Open your terminal and run the following command to clone the repository:

```
git clone https://github.com/DaliborBilicky/location_sensitivity.git
```

This will create a folder named `location_sensitivity` containing the project files. Navigate to the project folder:

```
cd location_sensitivity
```

From here, you can follow the steps to install the required libraries and set up the environment for running the application.

Installing libraries

Before running the application, you'll need to install the required libraries. I recommend setting up a virtual environment to manage dependencies easily.

Creating a virtual environment (optional but recommended)

1. Create a virtual environment (you can name it `venv` or any name you prefer):

    ```
    python3 -m venv venv
    ```

2. Activate the virtual environment:

   - On **Windows**:

     ```
     .\venv\Scripts\activate
     ```

   - On **macOS/Linux**:

     ```
     source venv/bin/activate
     ```

   Once the virtual environment is activated, your terminal prompt will change, indicating that you're working inside the virtual environment.

Installing required libraries

With the virtual environment activated, install the required Python libraries using the `requirements.txt` file:

```
pip install -r requirements.txt
```

This will automatically install all necessary dependencies for the project.

Note: You can always check which libraries are installed by running:

```
pip freeze
```

If you encounter any issues during installation, make sure that you have the necessary Python version installed and that your pip is up-to-date:

```
python -m pip install --upgrade pip
```

After completing these steps, your environment will be ready to run the application.

This installation guide is based on instructions from Python's official tutorial on venv.

Usage

To start the application, open your terminal and run the following command:

```
python src/main.py <option> <region acronym> <P>
```

Arguments

- `<option>`
  Type of experiment to run:
  - `A` – Tests how the optimal solution changes as the sensitivity parameter `k` increases.
  - `F` – Finds the first value of `k` where the optimal solution changes.
- `<region acronym>`
  Region to run the experiment on. Choose one of the following Slovak region acronyms:
  - `BA` – Bratislava
  - `TT` – Trnava
  - `NR` – Nitra
  - `TN` – Trenčín
  - `ZA` – Žilina
  - `BB` – Banská Bystrica
  - `PO` – Prešov
  - `KE` – Košice
- `<P>`
  The number of facilities to locate (e.g., ambulance or fire stations).

Example

```
python src/main.py A ZA 12
```

Output

All result files are saved in the `./results/` directory. The output is stored in files with the following names:

- `<region acronym>-<P>-calculate-all-ks.txt`

- `<region acronym>-<P>-calculate-first-k.txt`

Each file contains comprehensive statistics for a given region and sensitivity parameter `k`. The output includes:

- **Sensitivity parameter `k`** and its **upper limit**
- **Objective value**
- **Selected p-medians** (indices of optimal facility locations)
- **Speed statistics**:
  - Speed of ambulance
  - Minimum / Maximum / Average / Most frequent speed decline (km/h)
- **Top 10 smallest and largest speed declines** with details on:
  - Edge (node-node pair)
  - Original and elongated edge cost
- **Edges connected to selected p-medians** with their respective speed declines

Example snippet:

```
 k: 6.2422, upper limit: 44.3887
Objective value: 16456.6566
Weighted p-medians:
[6, 15, 18, 23, 33, 44, 49, 51, 77, 83, 84, 85]
Speed of ambulance: 110
Min speed decline: 0.8086
Max speed decline: 15.5482
Average speed decline: 3.0817
Most often speed decline: 2.2976
```

Each run produces similar structured statistics, either:

- For a **single** value of `k`, or
- For **multiple values** in a loop (e.g. all `k` from 0 to the upper bound)

This makes the results easily traceable, comparable, and ready for further analysis or visualization.