Genetic p-Median Solver > Download Here!

Location-allocation problems typical deal with provisioning of resources between facilities based on historic demand. The p-median approach is one such model that aims to minimise the total demand-weighted distance between the demand points and the facilities. This NP-Hard problem aims to locate '*p*' facilities to serve '*n*' demand, by minimising the total demand-weighted distance between the facilities and the demand.

The Track&Know p-median tool uses a genetic algorithm approach to solve the problem in polynomial time. This tool plays an important role in translating mobility information into policy and management recommendations.

Implementation Details

This tool is a parallelised and containerised implementation in python of a Genetic Algorithm approach to solve the p-Median problem. The underlying model is based on the following research paper:

Alp, O., Erkut, E., & Drezner, Z. (2003). An efficient genetic algorithm for the p-median problem. Annals of Operations research, 122(1-4), 21-42.

The python model is served as a Django application with an Angular front-end, all wrapped in a docker container. Due to the complexity of the model the Celery is included in the application stack to make sure calculations can run asynchronously and to retain previous results if the container restarts.

To calculate the costs of journeys from a demand point to a p-facility this tool relies on access to Open Route Service. Further implementation information is available in the docs folder on Github.

Input Data

Data can be loaded into the container through the form on the web front end. The tool only requires a csv with each row providing a latitude,longitude pair for each demand point. Post simulation there is a download option to extract the data from the container.

Quick Start

If you have your own Open Route Service pass it using the ORS_HOST environment variable:

- `docker run -p 8000:8000 -e ORS_HOST=https://yourhost/ors ibadkureshi/tnk-pmed:latest`

If you are using the public Open Route Service (the API playground) pass you key to the code

- `docker run -p 8000:8000 -e ORS_KEY=your key ibadkureshi/tnk-pmed:latest`
  # this is not recommended due to rate limits and the code doesnt optimise against number of api/routing calls

Then open a browser and go to http://localhost:8000/

>> More Info