Programming Project I

An Analysis Tool for Railway Network Management

DA 2023 Instructors Team

Departamento de Engenharia Informática (DEI)/Departamento de Ciências de Computadores (DCC)
Faculdade de Eng. da Univ. do Porto (FEUP)/Faculdade de Ciências da Univ. do Porto (FCUP)

Spring 2023

Due Date: April 7, 2023 at midnight (PT time)

1. Objectives

This first programming assignment aims at exposing you to realistic implementation of algorithmic solutions presented in class and in particular to the greedy algorithmic approach in the context of maximum-flow problems. In addition, you will be required to work in teams of 2 or 3 students (3 is preferred) and hence will have to develop inter-personal and project management skills. This first programming project is not very complex, but still you are required at the end to prepare a short demo of the functionalities and interfaces you have developed. It is also an objective of this project to prepare you for short presentations where you will have to be succinct and time-conscientious, focusing on what it essential and what it is not.

This document describes the motivation of this project, the expected interface followed by a description of the problem statement and description of the demo you are expected to present. The problem statement includes a description of each task (alongside the corresponding grading). Lastly, we provide specific turn-in instructions you need to follow. **Recall, the deadline is April 7, 2023 at midnight.**

2. Problem Motivation

In this first programming project assignment, you are asked to implement an analysis tool that can support the management team of a railway network to make informed decisions about how to best allocate its resources, both financial as well as physical (e.g., trains). This tool, which will use a realistic data set, will allow management to decide on aspects such as how to best assign selected trains to lines, how to maximize its revenue when multiple levels of service are offered, or even, identify more sensitive sections of its network to failures so as to anticipate service disruption or at least to mitigate its nefarious effects.

3. Problem Data and Interface

To make your problem realistic, you are given a real dataset describing the Portuguese continental railway network. The dataset contains the indication of existing rail lines, stations and stops (along with their location on each line) and the segments between stations and/or stops. This dataset is described in the *Project1Data.zip* file annexed to this project description and whose illustration is depicted in Figure 1 below.

[T1.1: 1.0 point] Obviously, a first task will be to create a simple interface menu exposing all the functionalities implemented in the most user-friendly way possible. This menu will also be instrumental for you to showcase the work you have developed in a short demo to be held at the end of the project.

[T1.2: 1.0 point] Similarly, you will also have to develop some basic functionality (accessible through your menu) to read and parse the provided data set files. This functionality will enable you (and the eventual user) to select alternative railway networks for analysis. With the extracted information, you

are to create one (or more) appropriate graphs upon which you will carry out the requested tasks. The modelling of the graph is entirely up to you, so long as it is a sensible representation of the railway network and enables the correct application of the required algorithms.

[T1.3: 2.0 points] In addition, you should also include documentation of all the implemented code, using Doxygen, indicating for each implemented algorithm the corresponding time complexity.

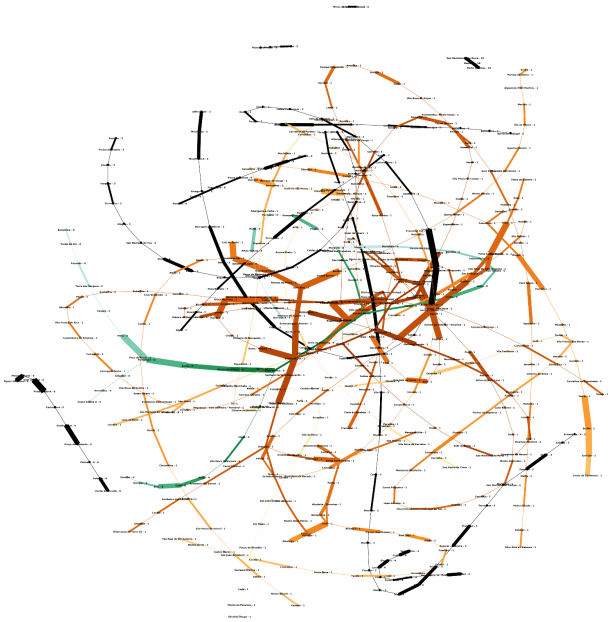

Figure 1. Illustration of the railway network described in the dataset file *Project1Data.zip*.

4. Problem Statement and Organization of Work

To facilitate your work, we have structured the functionalities you are expected to develop into three sets as follows.

4.1. Basic Service Metrics

The management team wishes to analyze the number of trains that it should purchase in order to serve the railway network fully. To support their needs, you should implement efficient algorithms to help them in responding to the following company requirements:

[T2.1: 3.5 points] Calculate the maximum number of trains that can simultaneously travel between two specific stations. Note that your implementation should take any valid source and destination stations as input;

[T2.2: 2.0 points] Determine, from all pairs of stations, which ones (if more than one) require the most amount of trains when taking full advantage of the existing network capacity;

[T2.3: 1.5 points] Indicate where management should assign larger budgets for the purchasing and maintenance of trains. That is, your implementation should be able to report the top-k municipalities and districts, regarding their transportation needs;

[T2.4: 1 point] Report the maximum number of trains that can simultaneously arrive at a given station, taking into consideration the entire railway grid.

4.2. Operation Cost Optimization

Management has settled on two types of services to be made available to the public: STANDARD and ALFA (PENDULAR). Each service requires different trains and has distinct train connections, which results in different costs for the company. For the purposes of your work, you can consider that the STANDARD service in a segment of the line has a fixed cost of 2€ per train while the ALFA service will cost the company 4€ per train in the same segment. Note that this is the cost of operation for the company, not the fare price for the customers, and that this cost is per train and per segment. Naturally, management would like to understand how to allocate trains to each segment in order to minimize their costs, while maintaining the same level of service. To this effect, you should provide an "optimization" analysis that takes these restrictions into consideration:

> **[T3.1: 2.0 points]** Calculate the maximum amount of trains that can simultaneously travel between two specific stations with minimum cost for the company. Note that your system should also take any valid source and destination stations as input;

4.3. Reliability and Sensitivity to Line Failures

To enhance the reliability of the railway network, management needs to make periodic maintenance and occasional repairs. This is a complex activity and one of its key aspect is forecasting of failures and repair needs. To help management, you are to implement a functionality to assess the sensitivity of some segments to failures as they may occasionally be unavailable due to a wide number of reasons (repairs, services, disasters, etc.)/ Specifically, your implementation should provide the following data:

> **[T4.1: 2.5 points]** Calculate the maximum number of trains that can simultaneously travel between two specific stations in a network of reduced connectivity. Reduced connectivity is understood as being a **subgraph** (generated by your system) of the original railway network. Note that your system should also take any valid source and destination stations as input;

> **[T4.2: 1.5 points]** Provide a report on the stations that are the most affected by each segment failure, i.e., the top-k most affected stations for each segment to be considered.

5. Demo & Presentation

Giving a presentation that is "short-and-to-the point" is increasingly important. As such, you are required to structure a short 15-minutes demo of your work, where you can highlight some of the aspects of your implementation, specifically:

> **[T5.1: 2.0 points]** Use the (hopefully) user-friendly interface you have developed to illustrate the various algorithm results for a sample set of railway grids which you should develop specifically for the purposes of this demo. For instance, you can develop a small set of very modest railway networks for contrived capacities so that you can highlight the "correctness" of your solution. For instance, a grid that has a "constricted" segment where all traffic must go through, will clearly have a segment very "sensitive" to failures.

Note that, being the "contrived" or "toy" grids developed for purposes of the demo, you are welcome to develop them in partnership with other groups, in this way, effectively cooperatively developing a large pool of small tests cases. In our evaluation, however, we will use other larger railway network input cases.

6. The Data Set

The provided dataset file (*Project1Data.zip*) includes two files, namely, **stations.csv** and **network.csv** described next.

The file named **stations.csv**, lists a description of each railway network station and stop. This description, of a total of 532 stations and stops) consists of the designation of the station/stop ("Name"), the district ("District"), municipality ("Municipality") and township ("Township") it belongs to and the main line ("Line") that it serves. A sample of a section of this description is illustrated below.

stations.csv

Name	District	Municipality	Township	Line
Porto Campanhã	Porto	Porto	Porto	Linha do Norte
Viana do Castelo	Viana do Castelo	Viana do Castelo	Viana do Castelo	Linha do Minho
Ermidas-Sado	Setúbal	Santiago do Cacém	Ermidas-Sado	Linha do Sul
Faro	Faro	Faro	Faro	Linha do Algarve
…	…	…	…	…

The file named **network.csv**, lists a description of the network segments (for a total of 515 segments) where each row describes a segment between 2 train stations ("Station_A" and "Station_B"). Stations are referred by their (unique) name ("Name"). The capacity ("Capacity") indicates the number of trains that can simultaneously travel on the respective network segment. The last column describes the type of service that is provided ("Service"), which can either be STANDARD or ALFA PENDULAR. A sample of a section of this description is illustrated below.

network.csv

Station_A	Station_B	Capacity	Service
Casa Branca	Monte das Flores	8	STANDARD
Monte das Flores	Évora	8	STANDARD
Évora	Portalegre	10	STANDARD
Funcheira	Santa Clara-Sabóia	4	ALFA PENDULAR
…	…	…	…

7. Turn-In Instructions & Deadline

Submit a zip file named DA2023_PRJ1_G<GN>.zip on Moodle, where GN stands for your group number, with the following content:

- Code folder (contains program source code)
- Documentation folder (contains html documentation, generated using Doxygen)
- Presentation file (PDF format) that will serve as a basis for the demonstration.

Late submissions, up to 24 hours and 48 hours, will incur a penalty of 10% and 25% of the grade, respectively. No submissions will be accepted 48 hours after the deadline. Exceptions apply for justified and documented technical submissions issues. **Recall, the deadline is April 7, 2023 at midnight.**