Practice 4 – Backtracking y Branch-and-Bound. A very efficient delivery man

Data Structures and Algorithms II. University of Almeria

Version **curso.2022**

Goals

- Construct solutions to a problem using the ***backtracking*** **and** ***branch-and-bound*** **algorithmic methods**. Analyze and compare the implemented algorithms from diferent perspectives.
- Carry out an **analysis of the efficiency** of the solutions provided, and a comparison both from a theoretical and practical point of view.

Requirements

To pass this practice, the following must be done:

- Remember (from EDA I) the proper **data structures** for problem solving.
- Know how to implement the ***backtracking*** **and** ***branch-and-bound*** **programming schemes** to respond to a specific need (in this case, solving the Traveling Salesman Problem problem, TSP, and Hamiltonian cycles problem).
- **Evaluate the implemented** ***backtracking*** **and** ***branch-and-bound*** **programming algorithms**, relating both algoritmic methods.

Problem statement

As we know, in EDAland there are very varied characters, among them, we find 'Braulio the delivery man', a tireless and responsible worker from Almeria who has spent a lifetime distributing Hoy-Almeria newspapers throughout the country. Every morning, when the publisher of Hoy-Almeria closes the daily edition and the first copies come out of the printing press, Braulio is there, with his van, ready to deliver them to all the cities of EDAland. In addition to his efficiency in distributing, Braulio also stands out for being a person who takes his economy very much into account, and always tries to choose the circuit that involves him traveling the shortest possible distance or spending the least amount of fuel (and even more so now, when high price that is) possible. To be so effective, Braulio raised three important questions. The first consisted of knowing all the possible paths, starting from Almeria, which go through each city where a batch of newspapers must be left, exactly once and later returning to Almeria. The second question was to determine a circuit that would cover each city exactly once, starting from and returning to Almeria, and having covered the shortest possible distance in total. The third and last question was to determine a circuit that runs through each city exactly once, leaving and returning to Almeria and whose fuel consumption is the minimum possible.

In order to respond to the questions raised by Braulio there is the **new reduced network of roads** of EDAland (Figure 1). In EDAland a new general road plan was carried out and some new roads (in blue) were added to the previous network (practice 2). As we can see, it connects the large cities of the country (21 cities with 41 roads), along with the distance in kilometers between two cities. For example, the distance that connects Almeria with Granada is 173 kilometers. This network will help us to verify the correct operation of the *backtracking* and *branch-and-bound* algorithms developed.

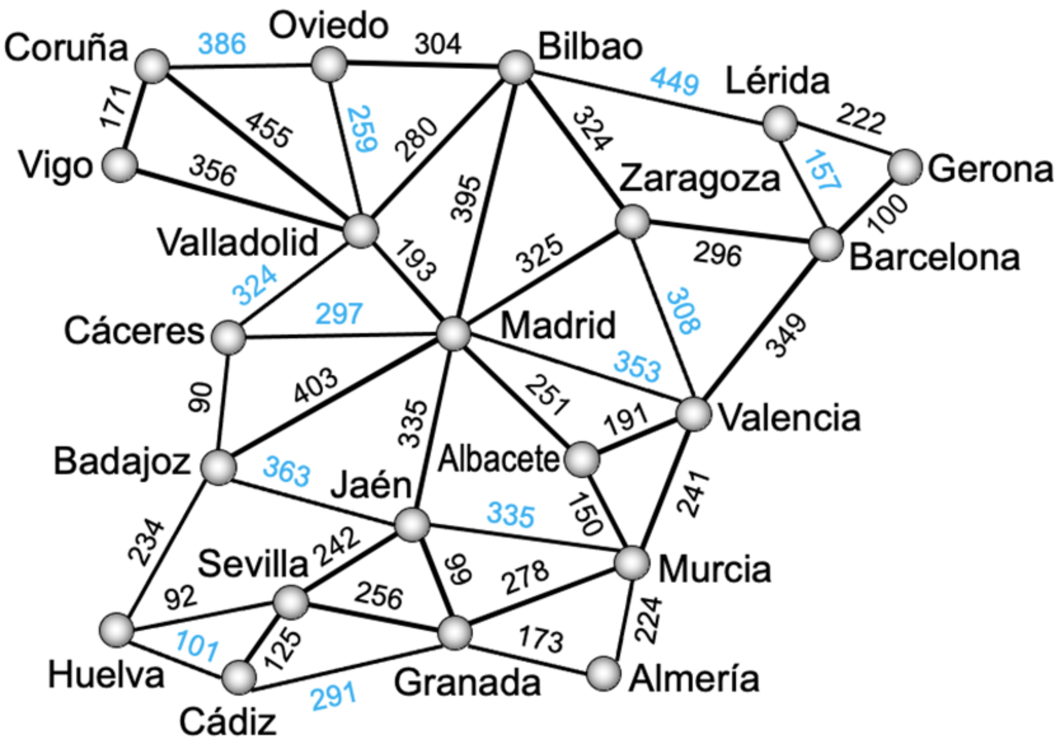

Figure 1. New reduced network of roads of EDAland

Once you check the correct operation of our algorithms in the new reduced network of roads, we will execute them in the **national network of roads** of EDALand (Figura 2, adapted from cities and roads graph published in [1]), that connects all population centers in the country (1053 population centers and 2017 roads with the distance between every two of them). That is the same network from practice 2.

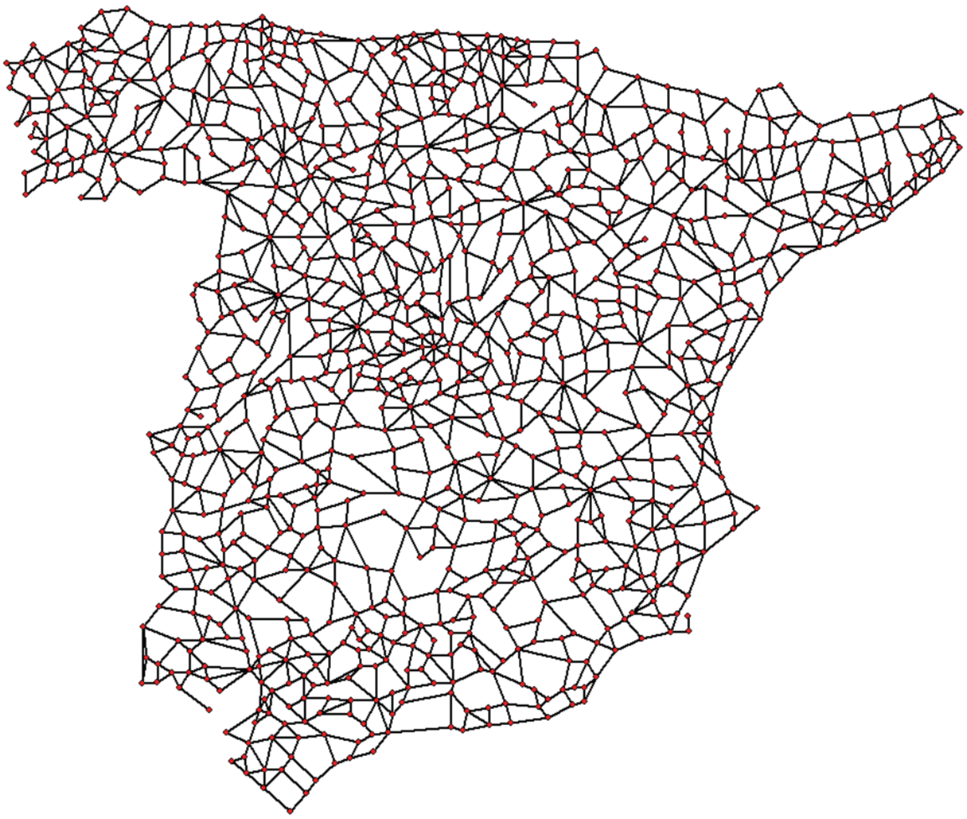

Figure 2. EDAland National Road Network

Work to develop

You must propose and implement algorithms solutions using ***backtracking*** and ***branch-and-bound***, as required, to the following problems:

- (***Backtracking***) Determine all the possible paths, if there were more than one, starting from Almeria, that go through each city of the new reduced road network of EDAland, where Braulio must leave a batch of newspapers, exactly once and return to Almeria.
- (***Backtracking***) Determine a path that, starting from Almeria, visits each city exactly once, returning to Almeria and having traveled the shortest distance. Solve this problem for the new reduced road network of EDAland.
- (***Branch-and-Bound***) Modify the value of the edges of the new reduced road network as follows (this assumes that we will have a new reduced road network with different values for the edges): multiply the distance value by a random real number *x*, *$0.5 \leq x \leq 1.0$*, so that the weight of the edges is now a real number (`double`) with two decimal places, which could correspond to the fuel cost of moving from one city to another. On this new network, that should be saved on a new datafile named `graphEDAlandTSPFuels.txt`, determine a path that Braulio must follow, starting from Almeria, going through each city exactly once, returning to Almeria and spending as little fuel as possible.
- Using the algorithms (***Backtracking*** and ***Branch-and-Bound***) implemented in previous sections, you must try to obtain the path in the complete national road network, starting from any urban nucleus, visiting each town exactly once, returning to the starting nucleus and having the shortest possible distance. What conclusions do you get from the attempt? Is

there any way to solve the problem that has been raised? Indicate everything in a reasonable way, justifying it correctly. In addition, to verify that the algorithms work on said network, carry out a trace of their execution in which it shows their status depending on the iteration or every certain time.

To do this, you must complete the following sections:

- **Study of the implementation**: Explain the most important details of the implementation, both of the data structures used to solve the specific problem, and of the implemented algorithms. The code must be reasonably well documented (`JavaDoc`).
- **Theoretical study**: Study the execution times of the implemented ***backtracking*** and ***branch-and-bound*** algorithms, depending on the number of population centers (vertexes) and the number of paths (edges). Also compare the proposed algorithms, taking into account the characteristics of the network (graph) and the chosen implementation techniques.
- **Experimental study**: Validation of the ***backtracking*** and ***branch-and-bound*** algorithms implemented on the EDAland networks provided. To do this, the execution times of the implemented algorithms must be obtained and compared. The theoretical and experimental results will be contrasted, checking if the experimental ones confirm the previously analyzed theoretical ones. The experiments carried out will be justified, and in case of discrepancy between the theory and the experiments, an attempt should be made to find a reasoned explanation.

Submissions

A private GitHub repository (same repository for all EDA II practices) with all the documentation and source code required in the practice must be submitted on date:

- In that repository, a new folder `practica_4`, with at least two subfolders, one for the documentation, `docs`, and one for the source code, `src`, folowing the same project structure explained in previous practice.
- A memory document that explains everything you have done in practice. The memory must have the format indicated below. If desired, you can also make a presentation of the practice.
- Source code of the application, developed in JAVA, which solves everything raised in practice. Remember that you will have to measure execution times of your solutions, so you must include the necessary commands for this in the source code.
- Test suite with unit tests that you consider appropriate to make sure that everything works properly (in the `test` folder).

The **memory** of this practice to deliver must be brief, clear and well written. This should include the following sections:

- A brief **introduction** with a theoretical study of the algorithmic method used in this practice (***backtracking*** and ***branch-and-bound***)..

- A section for each of the **proposed sections** to be developed in this practice (implementation study, theoretical study and experimental study). We must emphasize that the sections must be included in the same order in which they have been presented.
- An **annex** with the design of the implemented code will also be included, with a class diagram and any other diagram you consider useful, but do not include code here. In this annex, a list of the source files and a brief description of the content of each one must be included as well.
- It is important to always include the **bibliographical sources** used (web, books, articles, etc.) and refer to them in the document.

Assessment

Each section will be evaluated independently, although it is a necessary condition to pass the internship that the implemented programs work correctly.

- The implementation together with the documentation of the code will be valued out of 40%
- The study of the implementation will be valued out of 10%
- The theoretical study will be valued out of 15%
- The experimental study will be valued out of 35%

It will be penalized not deliver the theoretical introduction section or a bad presentation of the report.

The defense of the code and memory by the teacher may be required.

Deadline

Deadline: **June 5th 2022**

References

- [1] Gines García Mateos. The Traveler’s Challenge. Available online on http://dis.um.es/~ginesgm/retoviajante.html [Date of consult: 2022/03/19]