Final project - Travelling Salesman Problem

Fanoarii Boyer - Jérémy Beaugeard - Arthur Guerineau - Adrien Le Saux

CIR3 - Graph Theory - Leandro Montero

30 April 2020

Contents

Chapter 1

Introduction

In this final graph theory project, we try to solve the Travelling Salesman Problem (TSP) using different algorithms and heuristics. Before we get to the code, it is important to consider how we are going to model the graphs.

In the TSP, we work with undirected complete graphs, which means they are very dense. Adjacency lists are useful for sparse graphs, but in our case it will be more appropriate, in terms of complexity, to use adjacency matrix.

To implement the adjacency matrix we will use the C++ Boost library. In this project we seek to compare the performance between different solutions to the same problem. As C++ is a low-level programming language it is very fast and will be suitable for our use. Moreover, the boost library is very well known and widely documented.

Chapter 2

Real-life situations

Chapter 3

Exact algorithm

3.1 Pseudo-code

Data: this text
Result: how to write algorithm with LaTeX2e
while *not at end of this document* **do**
    read current;
    if *understand* **then**
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

3.2 Time complexity

3.3 Optimal Solution

3.4 Execution time and performance

Chapter 4

Constructive heuristic

4.1 Pseudo-code

4.2 Time complexity

4.3 Optimal Solution

4.4 Execution time and performance

Chapter 5

Local search heuristic

5.1 Pseudo-code

5.2 Time complexity

5.3 Optimal Solution

5.4 Execution time and performance

Chapter 6

GRASP meta-heuristic

6.1 Pseudo-code

6.2 Time complexity

6.3 Optimal Solution

6.4 Execution time and performance

Chapter 7

Conclusion

Bibliography