Discrete Optimization Project

INSA Rennes
Applied Mathematics Department
DMA06-OD - 2023

In this project, you will focus on the Travelling Salesman Problem (TSP). Given a set of cities and the distances between all pairs of cities, it consists in finding the shortest route that passes through each city once and only once. The TSP is is one of the most famous graph optimization problems and has been studied for decades. It is NP-hard and is known to become computationally challenging as the number of cities considered increases. When it comes to solving instances of the TSP, one can use either *exact* or *approximate* solution methods. The first class finds the optimal solution of the problem at the cost of a longer running time. The second class is usually faster but is not guaranteed to find the optimal solution to the problem.

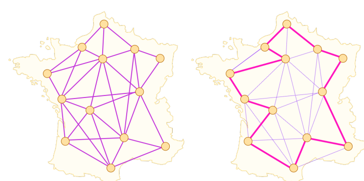

Figure 1: Shortest route through 13 different cities.

Objectives. You are asked to explore different strategies allowing to address TSP instances. More precisely, you will have to do the three following tasks.

1. Implement the **Held-Karp algorithm**, which exactly solves the TSP using a dynamic programming approach. This method was proposed by both Bellman [1] and Held and Karp [2] in 1962. It allows to solve small-size up to medium-size instances of the TSP.
2. Implement an exact solution method based on Integer Programming. You have to formulate the problem as an Integer Program and address it using a tailored solver. More precisely, you are asked to look at the **Miller–Tucker–Zemlin (MTZ)** and the **Dantzig–Fulkerson–Johnson (DFJ)** formulations of the problem [3, 4].
3. Implement an approximate solution method. You have the choice between the **Nearest Neighbor algorithm** [5], the **Lin–Kernighan heuristic** [6] or evolutionary algorithms such as the **Ant colony method** [7]. These methods are not guaranteed to solve the problem but they can provide good approximate solution within a reasonable time.

The different approaches will be presented during an oral evaluation. In particular, you may talk about complexity in time and space of the Held-Karp algorithm, discuss the advantages and flaws of the MTZ and the DFJ formulations or analyze the differences in running time and solution quality between exact and approximate solution methods. You goal is to show that you have a clear understanding of the underlying mechanisms used the different solution methods and a critical analysis on their implementation and practical relevance.

Contest. Groups that wish to do so can take part in a mini-competition. At the end of the project, 10 medium-size instance of the TSP will be randomly chosen. Each group can propose the algorithm of their choice to solve these instances. It is not mandatory to use one of the algorithms proposed in this project but the chosen method must be an exact one. The group with the fastest algorithm will be awarded with 2 bonus points.

Ressources. As a source of information, the Wikipedia pages of the TSP and of the solution methods are a good starting point. You are also advised to take a look at the papers aforementioned. For a more in-depth understanding of the subject, you can search for surveys on TSP solution methods on Google Scholar or read parts of one of the many books on the subject. The template for the code delivery as well as additional instructions are available at

`https://github.com/TheoGuyard/DMA06-OD`

Notation. A group providing an working implementation for the three different tasks starts with a score of at least 8/20. The code must run without throwing any error. The rest of the score will take into account the quality of the presentation, the level of understanding of the project and the clarity of the code delivered. The notation will also consider which approximate method the group has chosen to implement in the third task since some are more complex than others.

To go further. You may be able to complete the project before the deadline. In this case, you can go further. Any additional contributions to the project will be rewarded with bonus points.

References

[1] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*, 9(1):61–63, 1962.

[2] Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

[3] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

[4] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

[5] David S Johnson and Lyle A McGeoch. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1(1):215–310, 1997.

[6] Shen Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10):2245–2269, 1965.

[7] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.