ALGORITHM DESCRIPTION

NOTE: the main script has comments explaining the algorithm, also in this document all the support functions used in the script for loading data, visualizing it, etc… are not described.

Data Loading and Visualization

Inside the main() function of the script, the data for a directed graph is generated in three possible ways: loading the data from two CSV files, loading the data from one CSV file (comparison data) or generating a random directed graph.
After the data generation, the graph is visualized using Matplotlib (the script will continue closing the graph).
The three main data are: the points dict, the points profits dict and the arches dict with the cost associated

Model Creation

The Gurobi model is created and then the model variable, constraint and objective function are initialized.

Variables:

- Y_vertices = vector for selecting some or all of the given n vertices
- X_archs = matrix for selecting the arches for a vertex i to a vertex j

Constraints:

- Depot constraint = always selects the vertex 0
- Degree-2 constraint = every selected vertex (with the Y_vertices vector) needs to have one arch entering it and one arch leaving it, so a degree of two

Objective function:
The problem requires maximizing the total net profit: that is the difference between the total gains minus the total costs, where:

- total gains = sum of all the selected vertex profits
- total costs = sum of all the selected arches co

Model Optimization using lazy constraints

Using the lazy constraints functionalities of Gurobi, the model is optimized applying a sub-tour elimination for each step. The sub-tour elimination is considered only on the selected vertices of that step, and it is necessary to avoid sub-tours that don't consider the vertex 0 (because we can find solutions where the vertex 0 is selected, but it is inside another cycle!). The toy dataset shown in the default script, shows that the sub-tour elimination discard multiple independent cycles as a solution.

After the optimization, the results are shown printing the optimal path (as a list of selected vertices), the total maximum net profit obtained, the computation time and finally the graph tour is shown.