DATASETS:

The datasets used for testing the algorithm were two:

1) A **toy dataset**, containing an obvious solution and an invalid sub-tour with maximum gain but without the depot vertex 0 included in it

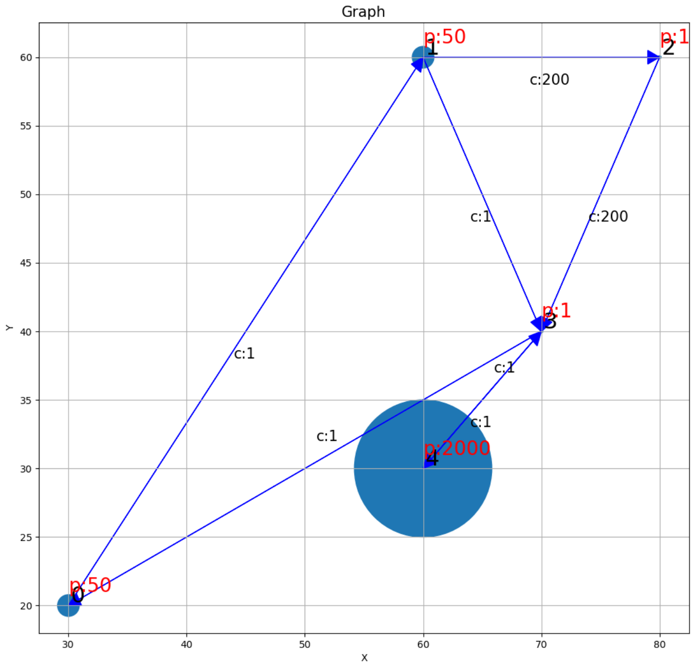
2) A **comparison dataset** used with other student to compare the optimization problem results

Additionally, other test were executed on random generated graph thanks to the built-in implemented functions.
Other implemented functions were made to also import dataset from a CSV file containing the graph data.

NOTE: all the dataset are present in CSV format inside the same directory of the main script

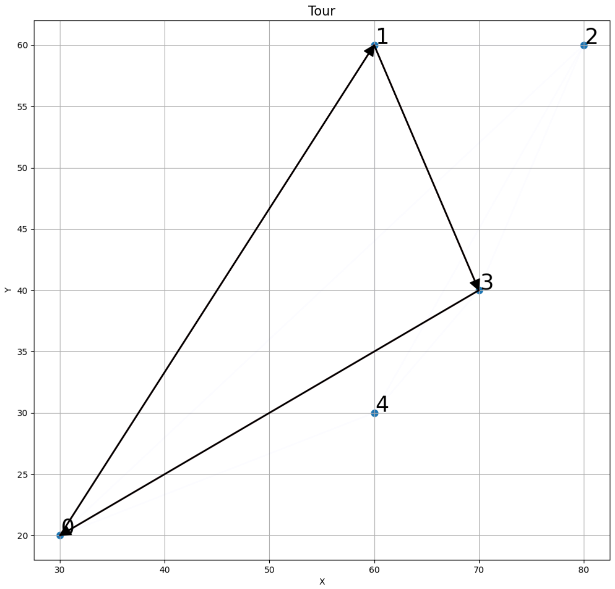
1) Toy dataset: “test_graph_points.csv” & “test_graph_costs.csv”

This dataset was manually created so to test a straightforward solution to the problem but with a test to the algorithm constraints: in this graph (see image below) it was possible to find an optimal solution selecting the sub-tour between the points 3 and 4, given the high gain of the vertex 4. Also, to find the correct solution, some arches needed to be discarded given their high costs.



The solution of this graph is the selection of nodes **0,1,3** with a total gain of 50 + 50 + 1 = 101 minus the arch cost 1 + 1 +1 = 3
The total profit is 101 - 3 = **98**

SOLUTION:
Points: {0: (30, 20), 1: (60, 60), 2: (80, 60), 3: (70, 40), 4: (60, 30)}
Optimal tour: **[0, 1, 3]**
Optimal cost: **98**
Time = 0.015625



In the left image: solution found for the directed graph.

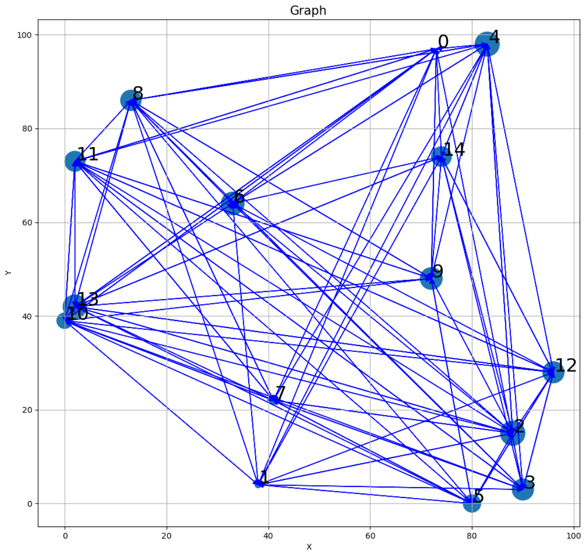
NOTE: This dataset is ready to be read and used in the script by default.

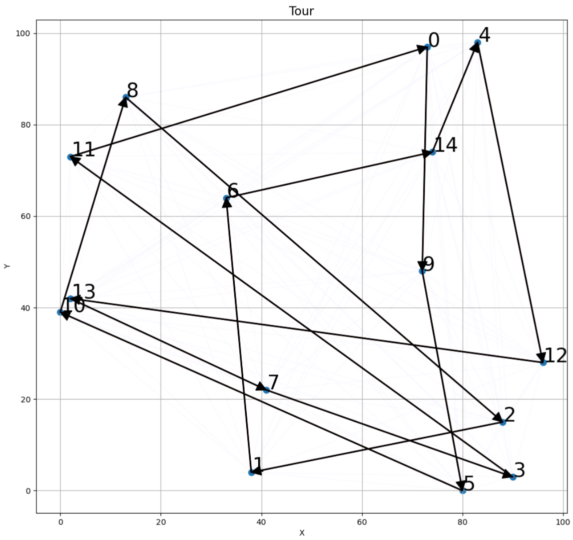
2) Comparison dataset: “dataset_1.csv”

This dataset was used with other student as a comparison metric. The result are identical to that of other two students who shared the same dataset.
Using other comparison datasets, there were changes in the selected path but the optimal cost obtained was similar.

SOLUTION:
Optimal tour: **[0, 9, 5, 10, 8, 2, 1, 6, 14, 4, 12, 13, 7, 3, 11]**
Optimal cost: **550**
Time = 0.046875

The images below shows the graph to test (left) and the results of the solution after the optimization (right)


Graph
X
Y


Tour
X
Y