# Multi-vehicle Set Orienteering Problem

This file is a short description of the project structure and files

## MSOP\Dataset\_Creation

* **dataset\_creation\_notebook.ipynb**: this is a jupyter notebook that reads the original sop instances from folder *SOP\_non\_random\_datasets* and creates the new multi-vehicle instance into folder *all*.
* The generated instance name is in the form of 217vm1084\_T60\_p1\_v4.msop
  + 217: number of sets
  + 1084: number of nodes
  + T60: 0.6\*optimal GTSP solution of instance 217vm1084
  + p1: profit category (p1: sum of cluster nodes, p2: high semi-random numbers)
* The generated instances need to be copied to the source code instances path *MSOP\MSOP\_datasets\all*

## MSOP\Visualisation

* **Running VisualizeSolution.py** and changing the dataset\_name = "217vm1084\_T60\_p1\_v4", an svg plot of the solution is exported.
* Instances may be visualized by **VisualizeDataset.py**.

## MSOP\MSOP

This is the main C# source code for the MSOP optimization.

Requirements:

* It requires installing Gurobi and adding a reference to **Gurobi100.NET.dll** from folder *\MSOP\MSOP\GRB*
* In order to use the LKH TSP algorithm of Helgaun, the files of *MSOP\MSOP\LKH* are required. These three files should be copied to the location of the executable (e.g., *\MSOP\MSOP\bin\Debug\netcoreapp3.1*).

The basic files that have the main functionality and parameters are the following:

* **Program.cs**: the program can be run by the Main function
* **Local\_Search.cs**: The Method GeneralLocalSearch is the optimization framework. It contains parameters and the scheme that it is used.
* **Subproblems.cs**: contains the current MIP that we use to simultaneously remove and insert sets to the routes by approximating the routing costs (best insertion cost)

Output:

* The optimization output is in folder *MSOP\MSOP\extracted\_solutions*