

Project ORTA 2024/2025

May 1, 2025

1 Description

You have been commissioned by the well-known poultry company *Polli Tech di N.I. & Co.* to design a distribution chain for their products. Specifically, the company requires you to find the optimal placement (among a set of possible locations) of warehouses, which are intended to distribute products to nearby supermarkets.

Each warehouse has a construction cost and can serve a specific subset of supermarkets. Any supermarket not served by at least one warehouse incurs an economic loss for the company. Finally, the placement of the warehouses must also consider the cost of transporting goods from the company to the warehouses, which is carried out by a single vehicle that departs from the company, visits each warehouse, and returns to the company every day.

2 Data

Each problem instance consists of the following files:

- `weights.json`: contains the costs the company may incur:
 - `'construction'`: daily cost due to construction and maintenance of a warehouse (we assume that the construction cost is not paid all at once but amortized over time)
 - `'missed_supermarket'`: daily penalty for each supermarket not served by any warehouse
 - `'travel'`: fuel cost per kilometer traveled

- **service.csv**: a matrix in which each row refers to a possible warehouse location and each column to a supermarket. If a warehouse can serve a given supermarket, the corresponding matrix element is 1, otherwise 0.
- **distances.csv**: a distance matrix between the possible warehouse locations and between these locations and the company. Both the rows and columns include the company as the first element, followed by the warehouses in the same order as in **service.csv**. Each matrix element (not necessarily symmetric) represents the distance in kilometers from the row location to the column location.

3 Requirement

Using Python as the programming language and GUROBI as the solver, develop a linear programming model to solve the problem.

4 Technical Requirements

The project must be completed in groups of 1 to 3 students.

The solver must consist of a single file named:

`solver_XXXXXX_YYYYYY_ZZZZZZ.py`

where XXXXXX, YYYYYY, and ZZZZZZ are the student IDs of the group members. For example, if the group consists of two students with IDs 123456 and 654321, the file should be named:

`solver_123456_654321.py`

If student 999999 decides to complete the project individually, the file should be named:

`solver_999999.py`

This file must contain a class that inherits from **AbstractSolver** and must have the **exact same name** as the file. For example, the file:

`solver_123456_654321.py`

must contain the class:

```
class solver_123456_654321(AbstractSolver):
```

Within this class, you must implement the method `solve()`, which takes no input (as all data is available in `self.inst`) and must return, in order, the vector `X` and the matrix `Y`.

- The vector `X` is a binary vector with length equal to the number of possible warehouse locations. Each element is set to `1` (or `True`) if a warehouse is built in that location, `0` (or `False`) otherwise.
- The matrix `Y` is a binary matrix with dimensions equal to the number of warehouse locations plus one (same size as `distances.csv`). The first row and column refer to the company, and the remaining ones refer to the warehouse locations. Each element `i,j` is equal to `1` (or `True`) if the vehicle's route includes the arc from `i` to `j`, `0` (or `False`) otherwise.

5 Testing

To test the solver, follow these steps:

- Place the file inside the `'solvers'` folder;
- Update `'solvers/__init__.py'` by importing the solver and adding the class name to `__all__`;
- Replace `DummySolver` in `'main.py'` with your solver;
- Run `'main.py'` to use the solver;
- Run `'evaluator.py'` to evaluate the total cost of the solution found by your solver.

6 Important Note

The submitted solvers will be tested exactly as described above, using the **same tools** made available during development. Therefore, if the files `main.py` or `evaluator.py` throw an error (due to file name, class name,

incorrect output format, or any other issue), the project will be automatically assigned a grade of **0 points**. The code will be checked on the structure that can be founded [here](#).

7 Deadline

The project must be submitted through the **Elaborati** section of **Portale della didattica** (one submission per group) by **23:59 on 30/06/2025**.