

Projet de Recherche Opérationnelle — Rapport

1. Introduction

Ce projet consiste à développer une application permettant de modéliser et résoudre un problème de **localisation-allocation** : décider où implanter des centres de distribution et déterminer quels clients seront desservis par quelles installations.

L'application est développée en **Python**, avec une IHM en **PySide6**, et utilise **Gurobi** pour résoudre des modèles de Programmation Linéaire (PL), Linéaire en Nombres Entiers (PLNE), et Mixte (PLM).

2. Objectifs du Projet

- Fournir une **interface graphique intuitive** pour saisir les données, lancer le solveur et visualiser les résultats.
- Modéliser précisément un problème d'optimisation réel.
- Résoudre le modèle avec Gurobi de manière **non bloquante** grâce au multithreading.
- Afficher, analyser et interpréter les solutions obtenues.

3. Modélisation du Problème

Variables de décision

- $y_j = 1$ si l'installation j est ouverte, 0 sinon.
- x_{ij} = quantité du client i desservie par l'installation j .

Fonction objectif

Minimiser

$$\sum_j f_j y_j + \sum_{i,j} c_{ij} x_{ij}$$

où :

- f_j est le coût d'ouverture,
- c_{ij} le coût de transport (dépendant de la distance).

Contraintes

1. Satisfaction de la demande :

$$\sum_j x_{ij} = d_i \quad \forall i$$

2. Capacité des installations :

$$\sum_i x_{ij} \leq Cap_j \cdot y_j \quad \forall j$$

3. Domaines :

- $x_{ij} \geq 0$
- $y_j \in \{0,1\}$ (PLNE) ou $y_j \in [0,1]$ (PL)

Ce modèle correspond au **Facility Location Problem (FLP)**, un classique des systèmes logistiques.

4. Développement de l'IHM

L'application est construite en **PySide6** et intègre :

4.1 Saisie des données

- Tables dynamiques (QTableWidget) pour :
 - les installations (capacités, coûts fixes, coordonnées)
 - les clients (demandes, coordonnées)

4.2 Résolution non bloquante

- Utilisation de **QThread** pour exécuter Gurobi sans bloquer l'IHM.
- Indicateur de progression pendant le calcul.

4.3 Visualisation

- Matplotlib intégré dans l'interface :
 - points représentant installations et clients,

- lignes indiquant l'allocation optimale,
- affichage du coût total et des décisions y_j, x_{ij} .

5. Résolution et Tests

Plusieurs jeux de données ont été testés pour vérifier :

- la cohérence des contraintes,
- la stabilité du solveur,
- le comportement de l'IHM sous forte charge (PLNE avec beaucoup de variables).

Les résultats fournis par Gurobi sont systématiquement cohérents :

- les installations non rentables restent fermées,
- les clients sont affectés à la solution minimisant les coûts,
- les capacités sont strictement respectées.

6. Analyse des Résultats

L'analyse montre que :

- l'ouverture d'une installation n'est justifiée que si les économies de transport compensent son coût fixe,
- les clients proches d'une même installation ont tendance à être regroupés,
- les contraintes de capacité influencent fortement la structure des allocations.

Le modèle peut facilement être enrichi pour intégrer des éléments complexes : fenêtres de temps, multi-capacité (volume/poids), multi-objectifs, etc.

7. Conclusion

Ce projet combine **modélisation mathématique**, **programmation Python**, **IHM professionnelle**, et **optimisation avancée** via Gurobi.

L'application obtenue est un outil complet permettant de :

- saisir des données logistiques,

- modéliser un problème de localisation–allocation,
- résoudre efficacement des PL / PLNE / PLM,
- visualiser et interpréter les solutions.

Elle constitue une base solide pour étendre le modèle à des cas industriels réels.