Optimisation de l'Emplacement de Panneaux Publicitaires

Modélisation Mathématique et Résolution via Gurobi

1. Contexte et Objectif

Dans le cadre d'une stratégie marketing, ce projet vise à déterminer les **meilleures localisations** pour implanter un nombre fixe de panneaux publicitaires (k). L'objectif est de maximiser l'audience totale (population pondérée) située dans un rayon de visibilité R autour des panneaux choisis. Ce problème est une instance classique du **Problème de la Couverture Maximale (Max-k-Coverage)**.

2. Modélisation Mathématique (PLNE Binaire)

Le problème est formulé comme un **Programme Linéaire en Nombres Entiers Binaire (PLNE-B)**.

Données et Paramètres :

- Ensemble des points de population (cibles).
- Ensemble des emplacements candidats potentiels.
- Poids (population/importance) du point cible
- Nombre exact de panneaux à installer (Budget).
- Matrice d'incidence (binaire) pré-calculée selon la distance :
  - a_{ij} = 1 si la distance d_{ij} =< R (le candidat j couvre i).
  - a_{ij} = 0 sinon.

Variables de Décision :

- Variable binaire égale à 1 si le candidat j est sélectionné, 0 sinon.
- Variable binaire égale à 1 si la population i est couverte, 0 sinon.

Formulation Complète :

$$
\begin{aligned}
\textbf{Maximiser (Z)} \quad & \sum_{i \in I} w_i \cdot y_i && \text{(Maximiser la population totale couverte)} \\
\textbf{Sujet à :} \quad & \sum_{j \in J} x_j = k && \text{(Contrainte de cardinalité : choisir k sites)} \\
& y_i \leq \sum_{j \in J} a_{ij} \cdot x_j \quad \forall i \in I && \text{(Un point n'est couvert que si un site proche est choisi)} \\
& x_j, y_i \in \{0, 1\} && \text{(Contraintes d'intégrité binaires)}
\end{aligned}
$$

3. Analyse du Modèle : Pourquoi un PLNE(B) ?

Ce modèle appartient à la classe des PLNE-B (*Binary Integer Linear Programming*) pour les raisons suivantes :

1. **Binaire (B) :** Toutes les décisions sont dichotomiques. On ne peut pas installer "0.5 panneau" et un habitant ne peut pas être "à moitié couvert".
2. **Linéaire (L) :** La fonction objectif et les contraintes sont des sommes pondérées du premier degré. Il n'y a aucun produit de variables ni exposants.
3. **Nombres Entiers (NE) :** L'espace de solution est discret. La relaxation continue n'aurait aucun sens physique.

4. Implémentation et Résolution (Gurobi)

La résolution est effectuée via l'optimiseur **Gurobi** en Python.

1. **Pré-traitement :** Utilisation d'un KDTree pour calculer efficacement la matrice de couverture a_{ij} (distances géométriques).
2. **Construction :** Traduction directe des équations ci-dessus via model.addVars (variables binaires) et model.addConstr.
3. **Algorithme :** Gurobi utilise la méthode du **Branch-and-Bound** (Séparation et Évaluation) combinée à des coupes (Cuts) pour explorer l'arbre de décision et garantir l'optimalité globale de la solution, contrairement aux heuristiques gloutonnes (Greedy) qui ne fournissent qu'une approximation.