Programmation Linéaire en Nombres Entiers
PLNE

M. Heriniaina Razafinjatovo

Objectifs du cours

- Comprendre ce qu'est un problème de PLNE.
- Savoir modéliser un problème réel sous forme de PLNE.
- Identifier les composantes essentielles : variables, objectif, contraintes.
- Introduire PuLP.

Définition de la PLNE

Programmation Linéaire en Nombres Entiers (PLNE)

C'est un problème d'optimisation linéaire dans lequel certaines ou toutes les variables de décision doivent prendre des **valeurs entières**.

Forme générale :

$$\begin{aligned} &\text{Maximiser ou Minimiser } c^T x \\ &\quad \text{sous contraintes } Ax \leq b \\ &\qquad x \in \mathbb{Z}^n \text{ (entier) ou } x \in \mathbb{R}^n \text{ (relaxé)} \end{aligned}$$

Composantes d'un modèle PLNE

1. **Variables de décision** (ex: nombre d'objets à produire)
2. **Fonction objectif** (ex: maximiser le profit)
3. **Contraintes linéaires** (ex: budget, capacité, ressources)
4. **Nature des variables** : entières, binaires (0 ou 1), continues

Exemple simple de PLNE

Problème : Maximiser $Z = 3x + 2y$

$$\begin{aligned} \text{avec } x + y &\leq 4 \\ 2x + y &\leq 5 \\ x, y &\in \mathbb{N} \end{aligned}$$

Exemple simple de PLNE

Problème : Maximiser $Z = 3x + 2y$

$$\begin{aligned} \text{avec } x + y &\leq 4 \\ 2x + y &\leq 5 \\ x, y &\in \mathbb{N} \end{aligned}$$

Interprétation : Choisir des valeurs entières de x et y satisfaisant les contraintes pour maximiser Z.

Méthodes de résolution

- **Relaxation continue** : on ignore l'aspect entier, on résout avec le simplexe.
- **Méthode de Branch and Bound** : exploration récursive de sous-problèmes.
- **Méthode de Branch and Cut** : ajout de contraintes coupantes.
- Utilisation de solveurs comme **CBC**, **GLPK**, **Gurobi**, etc.

PuLP

- PuLP est une bibliothèque Python permettant de formuler des problèmes d'optimisation linéaire sous forme symbolique.
- Elle ne résout pas directement : elle modélise le problème, puis appelle un solveur externe.
- Elle fonctionne très bien pour les problèmes en variables continues et les problèmes en variables entières (PLNE).

Exemple

Maximiser Z=3x+2y

sous contraintes :

$$\begin{cases} x + y \leq 4 \\ 2x + y \leq 5 \\ x, y \in \mathbb{N} \end{cases}$$

Syntaxe de base

```
from pulp import LpMaximize, LpProblem, LpVariable,
LpInteger

# 1. Définir le modèle d'optimisation avec un
objectif de maximisation
model = LpProblem(name="exemple-plne", sense=LpMaximize)


# 2. Définir les variables de décision : x et y (entières,
     non négatives)
x = LpVariable(name="x", lowBound=0, cat=LpInteger)
y = LpVariable(name="y", lowBound=0, cat=LpInteger)


# 3. Définir la fonction objectif
model += 3 * x + 2 * y, "Fonction_objectif"
```

Syntaxe de base

```
# 4. Ajouter les contraintes
model += (x + y <= 4, "Contrainte_1")
model += (2 * x + y <= 5, "Contrainte_2")

# 5. Résoudre le problème avec le solveur par défaut (CBC)
model.solve()

# 6. Afficher les résultats
print(f"Statut : {model.status}, {model.solver.status}")
print(f"x = {x.value()}, y = {y.value()}")
print(f"Valeur optimale de Z = {model.objective.value()}")
```

Remarques

- cat=LpInteger rend les variables entières.
- lowBound=0 impose qu'elles soient positives.
- model.solve() utilise CBC automatiquement.
-