Devoir Travaux Pratique 1 : Techniques d'Optimisation
Programmation linéaire

Nom: Selladurai Gowshigan

1 Introduction

Dans ce devoir, nous allons mettre en pratique les techniques d'optimisations qu'on a vu en cours. Nous allons résoudre des problèmes d'optimisations linéaires de manière graphique et utilisant python, plus spécifiquement à l'aide de la bibliothèque PuLP.

2 Formulation Mathématique du Problème

Pour chaque problème, nous définissons les variables de décision, la fonction objectif et les contraintes.

- Variables de décision :
  - $P1$: nombre de P1.
  - $P2$: nombre de P2.
  - $P3$: nombre de P3.
- Fonction objectif (à maximiser) :

$$Z = 60 * P1 + 50 * P2 + 70 * P3$$

- Contraintes :

$$\begin{aligned} 2 * P1 + P2 + 2 * P3 &\leq 100 && \text{(Contrainte de Machine 1)} \\ P1 + 2 * P2 + P3 &\leq 80 && \text{(Contrainte de Machine 1)} \\ P1 + P2 + 2 * P3 &\leq 90 && \text{(Contrainte de Machine 3)} \\ P1 \geq 0, P2 \geq 0, P3 &\geq 0 && \text{(Contrainte de Non Négativité)} \end{aligned}$$

3 Présentation du code python

```
#import pulp

# Création du problème de maximisation avec PuLP
prob = pulp.LpProblem("Maximiser_le_profit_de_l'entreprise", pulp.LpMaximize)

# Définition des variables de décision avec pour borne inférieure 0
et catégorie de variable ici entier car on peut pas avoir une partie de produit
P1 = pulp.LpVariable('P1', lowBound=0, cat='Integer') # Produit 1
P2 = pulp.LpVariable('P2', lowBound=0, cat='Integer') # Produit 2
P3 = pulp.LpVariable('P3', lowBound=0, cat='Integer') # Produit 3
```

```
# Fonction objectif => maximiser le profit en euros.
prob += 60*P1+50*P2+70*P3, "Profit_total"

# Contraintes:
prob += 2*P1 +P2+ 2*P3 <= 100, "Contrainte_Machine_1" # Production sur machine 1
prob += P1 +2*P2+ P3 <= 80, "Contrainte_Machine_2" # Production sur machine 2
prob += P1 +P2+ 2*P3 <= 90, "Contrainte_Machine_3" # Production sur machine 3

# Résolution du problème avec PuLP
prob.solve()

# Affichage des résultats
print("Statut de la solution :", pulp.LpStatus[prob.status])
print(f"Quantité optimale de P1 : {P1.varValue}")
print(f"Quantité optimale de P2 : {P2.varValue}")
print(f"Quantité optimale de P3 : {P3.varValue}")
print(f"Profit total optimal : {pulp.value(prob.objective)}")
```

4 Résultats Obtenus

Les résultats de la résolution du problème sont les suivants :

- Quantité optimale de P1 : 10
- Quantité optimale de P2 : 20
- Quantité optimale de P3 : 30
- Profit total optimal : 3700€

5 Analyse et Recommandations

Nous avons procédé à une formulation mathématique afin de bien définir le problème, puis nous avons implémenté un code python avec la bibliothèque PuLP pour résoudre notre problème d'optimisation linéaire.
Nous avons trouvé que le profit optimal est de 3700 avec pour quantité de P1:10, P2:20 et P3:30

En termes de remarques:
Nous pouvons constater qu'on a choisi un solveur parmi d'autres,on pourrait par exemple tester notre solutions sur d'autres modèles.

Nous pouvons également discuter avec l'entreprise pour voir si les contraintes sont fixes/flexible ou il existe d'autres qui sont pas précisé par l'entreprise afin d'optimiser encore plus le résultat.

Nous pouvons pousser l'analyse plus loin:

Nous pouvons discuter des contraintes qui limitent la production en terme de capacité quels machines augmenter, quels machines diminuer.

Nous pouvons discuter également de la rentabilité des produits avec l'entreprise et choisir de garder ou exclure certains si possible de la production.

6 Conclusion

En conclusion,ce travail pratique nous à permis de manipuler un problème concret afin de trouver une solution optimale à un problème d'optimisation linéaire afin de maximiser le profit grâce à python et la bibliothèque PuLP. Cela nous montre l'utilité et la puissance de type d'outils dans l'industrie.