

Recherche Opérationnelle

TP sur la résolution des problèmes de mélange et de planification de production

Les solveurs sont ceux de l'OR-Tools de Google dans l'environnement Python

Exercice 1

Développez et testez un modèle linéaire en nombres réels qui résout le problème de mélange sur la fabrication de l'acier vu en cours. Ecrivez un modèle compact en regroupant les contraintes et les variables avec des boucles FOR. La modélisation mathématique est donnée en couleur bleue dans le support du cours.

Indications :

```
max_mang = 1.65
Carb = [2.5, 3, 0, 0, 0, 0, 0]
X = []
for i in range(n):
    x = solver.NumVar(0, Stock[i], 'x_'+str(i))
    X.append(x)
ct_D = solver.Constraint(D, D, 'ct_D')
for i in range(n):
    ct_D.SetCoefficient(X[i], 1)
```

Exercice 2

Ecrivez un nouveau modèle pour le problème précédent en lisant toutes les données à partir d'un fichier Excel.

Indications :

```
import openpyxl as op
file = 'Interface_TP2_Exo2.xlsx'
wb = op.load_workbook(file)
ws = wb["input"]
min_carb = ws.cell(2,2).value
Cuiv = []
for i in range(2,n+2):
    Cuiv.append(ws.cell(i,8).value)
```

Exercice 3

Développez et testez un modèle linéaire en nombres réels qui résout le problème de planification de la production d'aliments pour bétail. Ecrivez un modèle compact avec des boucles FOR et une interface Excel.

Indications :

```
for i in range(aliment):
    U = []
    for j in range(mp):
        x = solver.NumVar(0.0, infinity, 'x_'+str(i)+'_'+str(j))
        U.append(x)
    X.append(U)
for i in range(aliment):
    ct_D = solver.Constraint(D[i], D[i], 'ct_D_'+str(i))
    for j in range(mp):
        ct_D.SetCoefficient(X[i][j], 1)
```