Projet recherche opérationnelle

Rayan Ayouaz (000429095)

Avril 2020

1 Introduction

En optimisation mathématique, un problème d'optimisation linéaire demande de minimiser une fonction linéaire sur un polyèdre convexe.
Dans le cadre du cours d'algorithmique et recherche opérationnelle, il nous est demandé de créer un programme linéaire et de l'implémenter au profit d'un industriel dans le but de l'aider à optimiser sa production.

2 Problème du menuisier

Chaque meuble requiert un ensemble de panneaux de longueur variable à découper dans des planches de longueur fixe. La largeur des panneaux et des planches coïncident. Comment minimiser le nombre de planche utilisées pour la découpe des panneaux ?

3 Implémentation

Nous nous servirons d'un script python qui prend en argument le chemin vers un fichier d'instance pour produire un fichier au format CPLEX qui pourra être exécuter par un solveur GLPK.

4 Modélisation linéaire

4.1 Constantes et données du problème

Considérons n panneaux à découper dans m planches. L'ensemble des panneaux de même taille ont une certaine longueur L_j et un certains nombre associé à produire N_j et les planches ont toutes une longueur L.

4.2 Ensembles

Nous considérons deux ensembles:

- $I = \{1, m\}$
  L'ensemble des planches.
- $J = \{1, n\}$
  L'ensemble des types de panneaux selon leur longueur.

4.3 Variables

- $x_{i,j}$: Le nombre de planche de type j découpées dans la planche i.
- y_i : La planche i a été utilisée.

La première variable est un nombre entier positif. La seconde est une variable binaire.

4.4 Programme linéaire

4.4.1 Modélisation sous contraintes

$$\min z = \sum_{i \in I} y_i$$

$$\text{s. t.} \begin{cases} \sum_{j \in J} x_{ij} * l_j \leq L * y_i & \forall i \in I & (a) \\ \sum_{i \in I} x_{ij} = N_j & \forall j \in J & (b) \\ x_{ij} \geq 0 & \forall i \in I, \forall j \in J & \\ y_i \in \{0, 1\} & \forall i \in I & \end{cases}$$

4.5 Explication des contraintes

Objectif: Minimiser le nombre de panneaux utilisés (la somme des variables binaire qui indiquent si un panneau est utilisé)

a Contrainte de longueur finie des planches
  La somme des panneaux de toutes les taille produit sur une planche multipliés par la longueur de ces panneaux, doit être inférieur ou égal à la longueur d'une planche, si elle est utilisée.

b Le bon nombre de panneaux sont produits
  La somme de tout les panneaux de type j produit sur chaque planche planche doit être égale au nombre de panneaux à produire du type j.

5 Fichier CPLEX

```

Minimize
  obj: y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8
Subject To
   board_length_1: 1.5 x_1_1 + 0.75 x_1_2 + 0.22 x_1_3 - 4.0 y_1 <= 0
   board_length_2: 1.5 x_2_1 + 0.75 x_2_2 + 0.22 x_2_3 - 4.0 y_2 <= 0
   board_length_3: 1.5 x_3_1 + 0.75 x_3_2 + 0.22 x_3_3 - 4.0 y_3 <= 0
   board_length_4: 1.5 x_4_1 + 0.75 x_4_2 + 0.22 x_4_3 - 4.0 y_4 <= 0
   board_length_5: 1.5 x_5_1 + 0.75 x_5_2 + 0.22 x_5_3 - 4.0 y_5 <= 0
   board_length_6: 1.5 x_6_1 + 0.75 x_6_2 + 0.22 x_6_3 - 4.0 y_6 <= 0
   board_length_7: 1.5 x_7_1 + 0.75 x_7_2 + 0.22 x_7_3 - 4.0 y_7 <= 0
   board_length_8: 1.5 x_8_1 + 0.75 x_8_2 + 0.22 x_8_3 - 4.0 y_8 <= 0
   all_panels_1:  x_1_1 +  x_2_1 +  x_3_1 +  x_4_1 +  x_5_1 +  x_6_1 +  x_7_1 +  x_8_1 = 4.0
   all_panels_2:  x_1_2 +  x_2_2 +  x_3_2 +  x_4_2 +  x_5_2 +  x_6_2 +  x_7_2 +  x_8_2 = 6.0
   all_panels_3:  x_1_3 +  x_2_3 +  x_3_3 +  x_4_3 +  x_5_3 +  x_6_3 +  x_7_3 +  x_8_3 = 12.0
Bounds
    x_1_1 >= 0
    x_1_2 >= 0
    x_1_3 >= 0
    x_2_1 >= 0
    x_2_2 >= 0
    x_2_3 >= 0
    x_3_1 >= 0
    x_3_2 >= 0
    x_3_3 >= 0
    x_4_1 >= 0
    x_4_2 >= 0
    x_4_3 >= 0
    x_5_1 >= 0
    x_5_2 >= 0
    x_5_3 >= 0
    x_6_1 >= 0
    x_6_2 >= 0
    x_6_3 >= 0
    x_7_1 >= 0
    x_7_2 >= 0
    x_7_3 >= 0
    x_8_1 >= 0
    x_8_2 >= 0
    x_8_3 >= 0
Integer
    x_1_1
    x_1_2
    x_1_3
```

```
    x_2_1
    x_2_2
    x_2_3
    x_3_1
    x_3_2
    x_3_3
    x_4_1
    x_4_2
    x_4_3
    x_5_1
    x_5_2
    x_5_3
    x_6_1
    x_6_2
    x_6_3
    x_7_1
    x_7_2
    x_7_3
    x_8_1
    x_8_2
    x_8_3
Binary
    y_1
    y_2
    y_3
    y_4
    y_5
    y_6
    y_7
    y_8
End
```

6 Fichier log

```
Problem:
Rows:        11
Columns:     32 (32 integer, 8 binary)
Non-zeros:   56
Status:      INTEGER OPTIMAL
Objective:   obj = 4 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 board_length_1
                                  -1                           0
     2 board_length_2
                                  -1                           0
     3 board_length_3
                               -0.25                           0
     4 board_length_4
                                   0                           0
     5 board_length_5
                                   0                           0
     6 board_length_6
                                   0                           0
```

```
     7 board_length_7
                                     0                             0
     8 board_length_8
                                 -0.61                             0
     9 all_panels_1                  4             4             =
    10 all_panels_2                  6             6             =
    11 all_panels_3                 12            12             =

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 y_1          *              1             0             1
     2 y_2          *              1             0             1
     3 y_3          *              1             0             1
     4 y_4          *              0             0             1
     5 y_5          *              0             0             1
     6 y_6          *              0             0             1
     7 y_7          *              0             0             1
     8 y_8          *              1             0             1
     9 x_1_1        *              2             0
    10 x_1_2        *              0             0
    11 x_1_3        *              0             0
    12 x_2_1        *              2             0
    13 x_2_2        *              0             0
    14 x_2_3        *              0             0
    15 x_3_1        *              0             0
    16 x_3_2        *              5             0
    17 x_3_3        *              0             0
    18 x_4_1        *              0             0
    19 x_4_2        *              0             0
    20 x_4_3        *              0             0
    21 x_5_1        *              0             0
    22 x_5_2        *              0             0
    23 x_5_3        *              0             0
    24 x_6_1        *              0             0
    25 x_6_2        *              0             0
    26 x_6_3        *              0             0
    27 x_7_1        *              0             0
    28 x_7_2        *              0             0
    29 x_7_3        *              0             0
    30 x_8_1        *              0             0
    31 x_8_2        *              1             0
    32 x_8_3        *             12             0

Integer feasibility conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

6.1 Lecture des résultats

Pour comprendre les résultats, nous pouvons observer combien de panneaux de type j sont fabriqué dans la planche i. Ce nombre est le résultat de la variables x_{ij}. Le numéro de la planche i n'ayant pas d'importance mais le nombre de planches utilisées aura bien été minimisé.

Pour cette instance nous utilisons à minima 4 planches. Dans la première planche on découpe 2 panneaux de taille 1.5. Dans la deuxième planche on découpe également 2 panneaux de taille 1.5. Dans la troisième planche on découpe 5 panneaux de longueurs 0.75 (longueur de la planche utilisé est alors égal à 3.75) . Finalement dans la dernière planche on découpe 1 panneau de taille 0.75 et 12 panneaux de taille 0.22 (longueur de la planche utilisé est alors égal à 3.39).

7 Conclusion

Les solveurs linéaire sont des outils très puissant. Néanmoins, l'importance d'une bonne modélisation est crucial si l'on veut réduire le nombre de variable produite par celle-ci. En effet, dans ce cas, le nombre de variables peut rapidement devenir important en fonction de l'instance considérée.