

Algorithmique et recherche opérationnelle

INFO-F310

Projet : CPLEX LP

26 avril 2020

Nom : BAKKALI Yahya

Matricule : 000445166

UNIVERSITÉ LIBRE DE BRUXELLES (ULB)

Table des matières

1	Introduction	2
2	Modèle	2
3	Script Python	3
4	Fichier CPLEX LP généré	3
5	Explication du fichier CPLEX LP généré	10
6	Log de GLPK obtenu	10
7	Explication du retour obtenu dans le fichier log	17
7.1	Problème	17
7.2	Fonction objectif I	18
7.3	Contrainte 1	18
7.4	Contrainte 2	19
7.5	Plan de découpe	20

1 Introduction

Pour ce projet, il est demandé d'écrire un script python qui, à partir d'un fichier d'instance indiquant un ensemble de panneaux de longueur variable à découper dans des planches de longueur fixe, génère une instance de programme linéaire en langage CPLEX LP qui détermine un plan de découpe de panneaux qui minimise le nombre de planches utilisées.

2 Modèle

Indices :

$M = \{1, \dots, m\}$ planches

$N = \{1, \dots, n\}$ panneaux

Constantes :

l_i = longueur du panneau i

L = longueur des planches j

Variables de décision :

$x_{i,j} \in \{0, 1\}$ telle que

$$x_{i,j} = \begin{cases} 1 & \text{si une planche } j \text{ est prise pour une découpe du panneau } i \\ 0 & \text{sinon} \end{cases}$$

$m_j \in \{0, 1\}$ telle que

$$m_j = \begin{cases} 1 & \text{si une planche } j \text{ est prise pour une découpe} \\ 0 & \text{sinon} \end{cases}$$

Formulation 1 Formulation linéaire

$$\min \quad \sum_{j \in M} m_j \quad (I)$$

$$\text{s.t.} \quad \sum_{i \in N} l_i x_{i,j} \leq L m_j \quad \forall j \in M \quad (1)$$

$$\sum_{j \in M} x_{i,j} = 1 \quad \forall i \in N \quad (2)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i \in N, \quad \forall j \in M \quad (3)$$

$$m_j \in \{0, 1\} \quad \forall j \in M \quad (4)$$

Les contraintes (1) assurent que si une planche est utilisée pour la découpe, sa longueur doit être supérieure ou égale à la somme des longueurs de tous les panneaux découpés depuis la même planche. D'autre part, (2) garantissent que chaque panneau a été correctement découpé et seulement à partir d'une des planches disponibles.

3 Script Python

Le script python `generate_lp_instance` est un script qui, en donnant en paramètre un chemin de fichier d'instance, génère le fichier CPLEX LP approprié qui modélise le problème.

Les lignes du fichier d'instance contiennent deux valeurs :

- un flottant représentant la longueur en mètres des planches pour la première ligne et des panneaux pour les lignes suivantes.
- un nombre entier correspondant au nombre de planches disponibles pour la première ligne et au nombre de panneaux à découper pour le reste des lignes.

En parcourant le fichier d'instance, quatre variables ont été créées M , N , L et l_i telles que :

M : un nombre entier représentant le nombre de planches disponibles.

N : un nombre entier représentant le nombre de panneaux à découper

L : un flottant représentant la longueur de toutes les planches

l_i : une liste contenant les longueurs de chaque panneau

Ces variables sont utilisées pour générer le fichier CPLEX LP adéquat.

4 Fichier CPLEX LP généré

```
Minimize
    obj: m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_8
Subject To
    c1_1: 4.0 m_1 - 1.5 x_1_1(1.5) - 1.5 x_2_1(1.5) - 1.5 x_3_1(1.5) - 1.5 x_4_1(1.5) - 0.75
           x_5_1(0.75) - 0.75 x_6_1(0.75) - 0.75 x_7_1(0.75) - 0.75 x_8_1(0.75) -
           0.75 x_10_1(0.75) - 0.22 x_11_1(0.22) - 0.22 x_12_1(0.22) - 0.22 x_13_1(0.22) - 0.22
           x_14_1(0.22) - 0.22 x_15_1(0.22) - 0.22 x_16_1(0.22) - 0.22 x_17_1(0.22) - 0.22
           x_18_1(0.22) - 0.22 x_19_1(0.22) - 0.22 x_20_1(0.22) - 0.22 x_21_1(0.22) - 0.22
           x_22_1(0.22) >= 0
    c1_2: 4.0 m_2 - 1.5 x_1_2(1.5) - 1.5 x_2_2(1.5) - 1.5 x_3_2(1.5) - 1.5 x_4_2(1.5) - 0.75
           x_5_2(0.75) - 0.75 x_6_2(0.75) - 0.75 x_7_2(0.75) - 0.75 x_8_2(0.75) - 0.75 x_9_2(0.75) -
```

$$\begin{aligned}
& 0.75 x_{10_2}(0.75) - 0.22 x_{11_2}(0.22) - 0.22 x_{12_2}(0.22) - 0.22 x_{13_2}(0.22) - 0.22 \\
& x_{14_2}(0.22) - 0.22 x_{15_2}(0.22) - 0.22 x_{16_2}(0.22) - 0.22 x_{17_2}(0.22) - 0.22 \\
& x_{18_2}(0.22) - 0.22 x_{19_2}(0.22) - 0.22 x_{20_2}(0.22) - 0.22 x_{21_2}(0.22) - 0.22 \\
& x_{22_2}(0.22) \geq 0 \\
\text{c1_3: } & 4.0 m_3 - 1.5 x_{1_3}(1.5) - 1.5 x_{2_3}(1.5) - 1.5 x_{3_3}(1.5) - 1.5 x_{4_3}(1.5) - 0.75 \\
& x_{5_3}(0.75) - 0.75 x_{6_3}(0.75) - 0.75 x_{7_3}(0.75) - 0.75 x_{8_3}(0.75) - 0.75 x_{9_3}(0.75) - \\
& 0.75 x_{10_3}(0.75) - 0.22 x_{11_3}(0.22) - 0.22 x_{12_3}(0.22) - 0.22 x_{13_3}(0.22) - 0.22 \\
& x_{14_3}(0.22) - 0.22 x_{15_3}(0.22) - 0.22 x_{16_3}(0.22) - 0.22 x_{17_3}(0.22) - 0.22 \\
& x_{18_3}(0.22) - 0.22 x_{19_3}(0.22) - 0.22 x_{20_3}(0.22) - 0.22 x_{21_3}(0.22) - 0.22 \\
& x_{22_3}(0.22) \geq 0 \\
\text{c1_4: } & 4.0 m_4 - 1.5 x_{1_4}(1.5) - 1.5 x_{2_4}(1.5) - 1.5 x_{3_4}(1.5) - 1.5 x_{4_4}(1.5) - 0.75 \\
& x_{5_4}(0.75) - 0.75 x_{6_4}(0.75) - 0.75 x_{7_4}(0.75) - 0.75 x_{8_4}(0.75) - 0.75 x_{9_4}(0.75) - \\
& 0.75 x_{10_4}(0.75) - 0.22 x_{11_4}(0.22) - 0.22 x_{12_4}(0.22) - 0.22 x_{13_4}(0.22) - 0.22 \\
& x_{14_4}(0.22) - 0.22 x_{15_4}(0.22) - 0.22 x_{16_4}(0.22) - 0.22 x_{17_4}(0.22) - 0.22 \\
& x_{18_4}(0.22) - 0.22 x_{19_4}(0.22) - 0.22 x_{20_4}(0.22) - 0.22 x_{21_4}(0.22) - 0.22 \\
& x_{22_4}(0.22) \geq 0 \\
\text{c1_5: } & 4.0 m_5 - 1.5 x_{1_5}(1.5) - 1.5 x_{2_5}(1.5) - 1.5 x_{3_5}(1.5) - 1.5 x_{4_5}(1.5) - 0.75 \\
& x_{5_5}(0.75) - 0.75 x_{6_5}(0.75) - 0.75 x_{7_5}(0.75) - 0.75 x_{8_5}(0.75) - 0.75 x_{9_5}(0.75) - \\
& 0.75 x_{10_5}(0.75) - 0.22 x_{11_5}(0.22) - 0.22 x_{12_5}(0.22) - 0.22 x_{13_5}(0.22) - 0.22 \\
& x_{14_5}(0.22) - 0.22 x_{15_5}(0.22) - 0.22 x_{16_5}(0.22) - 0.22 x_{17_5}(0.22) - 0.22 \\
& x_{18_5}(0.22) - 0.22 x_{19_5}(0.22) - 0.22 x_{20_5}(0.22) - 0.22 x_{21_5}(0.22) - 0.22 \\
& x_{22_5}(0.22) \geq 0 \\
\text{c1_6: } & 4.0 m_6 - 1.5 x_{1_6}(1.5) - 1.5 x_{2_6}(1.5) - 1.5 x_{3_6}(1.5) - 1.5 x_{4_6}(1.5) - 0.75 \\
& x_{5_6}(0.75) - 0.75 x_{6_6}(0.75) - 0.75 x_{7_6}(0.75) - 0.75 x_{8_6}(0.75) - 0.75 x_{9_6}(0.75) - \\
& 0.75 x_{10_6}(0.75) - 0.22 x_{11_6}(0.22) - 0.22 x_{12_6}(0.22) - 0.22 x_{13_6}(0.22) - 0.22 \\
& x_{14_6}(0.22) - 0.22 x_{15_6}(0.22) - 0.22 x_{16_6}(0.22) - 0.22 x_{17_6}(0.22) - 0.22 \\
& x_{18_6}(0.22) - 0.22 x_{19_6}(0.22) - 0.22 x_{20_6}(0.22) - 0.22 x_{21_6}(0.22) - 0.22 \\
& x_{22_6}(0.22) \geq 0 \\
\text{c1_7: } & 4.0 m_7 - 1.5 x_{1_7}(1.5) - 1.5 x_{2_7}(1.5) - 1.5 x_{3_7}(1.5) - 1.5 x_{4_7}(1.5) - 0.75 \\
& x_{5_7}(0.75) - 0.75 x_{6_7}(0.75) - 0.75 x_{7_7}(0.75) - 0.75 x_{8_7}(0.75) - 0.75 x_{9_7}(0.75) - \\
& 0.75 x_{10_7}(0.75) - 0.22 x_{11_7}(0.22) - 0.22 x_{12_7}(0.22) - 0.22 x_{13_7}(0.22) - 0.22 \\
& x_{14_7}(0.22) - 0.22 x_{15_7}(0.22) - 0.22 x_{16_7}(0.22) - 0.22 x_{17_7}(0.22) - 0.22 \\
& x_{18_7}(0.22) - 0.22 x_{19_7}(0.22) - 0.22 x_{20_7}(0.22) - 0.22 x_{21_7}(0.22) - 0.22 \\
& x_{22_7}(0.22) \geq 0 \\
\text{c1_8: } & 4.0 m_8 - 1.5 x_{1_8}(1.5) - 1.5 x_{2_8}(1.5) - 1.5 x_{3_8}(1.5) - 1.5 x_{4_8}(1.5) - 0.75 \\
& x_{5_8}(0.75) - 0.75 x_{6_8}(0.75) - 0.75 x_{7_8}(0.75) - 0.75 x_{8_8}(0.75) - 0.75 x_{9_8}(0.75) - \\
& 0.75 x_{10_8}(0.75) - 0.22 x_{11_8}(0.22) - 0.22 x_{12_8}(0.22) - 0.22 x_{13_8}(0.22) - 0.22 \\
& x_{14_8}(0.22) - 0.22 x_{15_8}(0.22) - 0.22 x_{16_8}(0.22) - 0.22 x_{17_8}(0.22) - 0.22 \\
& x_{18_8}(0.22) - 0.22 x_{19_8}(0.22) - 0.22 x_{20_8}(0.22) - 0.22 x_{21_8}(0.22) - 0.22 \\
& x_{22_8}(0.22) \geq 0 \\
\text{c2_1: } & x_{1_1}(1.5) + x_{1_2}(1.5) + x_{1_3}(1.5) + x_{1_4}(1.5) + x_{1_5}(1.5) + x_{1_6}(1.5) + \\
& x_{1_7}(1.5) + x_{1_8}(1.5) = 1
\end{aligned}$$

$$\begin{aligned}
c2_2: & x_2_1(1.5) + x_2_2(1.5) + x_2_3(1.5) + x_2_4(1.5) + x_2_5(1.5) + x_2_6(1.5) + \\
& x_2_7(1.5) + x_2_8(1.5) = 1 \\
c2_3: & x_3_1(1.5) + x_3_2(1.5) + x_3_3(1.5) + x_3_4(1.5) + x_3_5(1.5) + x_3_6(1.5) + \\
& x_3_7(1.5) + x_3_8(1.5) = 1 \\
c2_4: & x_4_1(1.5) + x_4_2(1.5) + x_4_3(1.5) + x_4_4(1.5) + x_4_5(1.5) + x_4_6(1.5) + \\
& x_4_7(1.5) + x_4_8(1.5) = 1 \\
c2_5: & x_5_1(0.75) + x_5_2(0.75) + x_5_3(0.75) + x_5_4(0.75) + x_5_5(0.75) + x_5_6(0.75) + \\
& x_5_7(0.75) + x_5_8(0.75) = 1 \\
c2_6: & x_6_1(0.75) + x_6_2(0.75) + x_6_3(0.75) + x_6_4(0.75) + x_6_5(0.75) + x_6_6(0.75) + \\
& x_6_7(0.75) + x_6_8(0.75) = 1 \\
c2_7: & x_7_1(0.75) + x_7_2(0.75) + x_7_3(0.75) + x_7_4(0.75) + x_7_5(0.75) + x_7_6(0.75) + \\
& x_7_7(0.75) + x_7_8(0.75) = 1 \\
c2_8: & x_8_1(0.75) + x_8_2(0.75) + x_8_3(0.75) + x_8_4(0.75) + x_8_5(0.75) + x_8_6(0.75) + \\
& x_8_7(0.75) + x_8_8(0.75) = 1 \\
c2_9: & x_9_1(0.75) + x_9_2(0.75) + x_9_3(0.75) + x_9_4(0.75) + x_9_5(0.75) + x_9_6(0.75) + \\
& x_9_7(0.75) + x_9_8(0.75) = 1 \\
c2_10: & x_10_1(0.75) + x_10_2(0.75) + x_10_3(0.75) + x_10_4(0.75) + x_10_5(0.75) + \\
& x_10_6(0.75) + x_10_7(0.75) + x_10_8(0.75) = 1 \\
c2_11: & x_11_1(0.22) + x_11_2(0.22) + x_11_3(0.22) + x_11_4(0.22) + x_11_5(0.22) + \\
& x_11_6(0.22) + x_11_7(0.22) + x_11_8(0.22) = 1 \\
c2_12: & x_12_1(0.22) + x_12_2(0.22) + x_12_3(0.22) + x_12_4(0.22) + x_12_5(0.22) + \\
& x_12_6(0.22) + x_12_7(0.22) + x_12_8(0.22) = 1 \\
c2_13: & x_13_1(0.22) + x_13_2(0.22) + x_13_3(0.22) + x_13_4(0.22) + x_13_5(0.22) + \\
& x_13_6(0.22) + x_13_7(0.22) + x_13_8(0.22) = 1 \\
c2_14: & x_14_1(0.22) + x_14_2(0.22) + x_14_3(0.22) + x_14_4(0.22) + x_14_5(0.22) + \\
& x_14_6(0.22) + x_14_7(0.22) + x_14_8(0.22) = 1 \\
c2_15: & x_15_1(0.22) + x_15_2(0.22) + x_15_3(0.22) + x_15_4(0.22) + x_15_5(0.22) + \\
& x_15_6(0.22) + x_15_7(0.22) + x_15_8(0.22) = 1 \\
c2_16: & x_16_1(0.22) + x_16_2(0.22) + x_16_3(0.22) + x_16_4(0.22) + x_16_5(0.22) + \\
& x_16_6(0.22) + x_16_7(0.22) + x_16_8(0.22) = 1 \\
c2_17: & x_17_1(0.22) + x_17_2(0.22) + x_17_3(0.22) + x_17_4(0.22) + x_17_5(0.22) + \\
& x_17_6(0.22) + x_17_7(0.22) + x_17_8(0.22) = 1 \\
c2_18: & x_18_1(0.22) + x_18_2(0.22) + x_18_3(0.22) + x_18_4(0.22) + x_18_5(0.22) + \\
& x_18_6(0.22) + x_18_7(0.22) + x_18_8(0.22) = 1 \\
c2_19: & x_19_1(0.22) + x_19_2(0.22) + x_19_3(0.22) + x_19_4(0.22) + x_19_5(0.22) + \\
& x_19_6(0.22) + x_19_7(0.22) + x_19_8(0.22) = 1 \\
c2_20: & x_20_1(0.22) + x_20_2(0.22) + x_20_3(0.22) + x_20_4(0.22) + x_20_5(0.22) + \\
& x_20_6(0.22) + x_20_7(0.22) + x_20_8(0.22) = 1 \\
c2_21: & x_21_1(0.22) + x_21_2(0.22) + x_21_3(0.22) + x_21_4(0.22) + x_21_5(0.22) + \\
& x_21_6(0.22) + x_21_7(0.22) + x_21_8(0.22) = 1 \\
c2_22: & x_22_1(0.22) + x_22_2(0.22) + x_22_3(0.22) + x_22_4(0.22) + x_22_5(0.22) + \\
& x_22_6(0.22) + x_22_7(0.22) + x_22_8(0.22) = 1
\end{aligned}$$

Binary

m_1
 m_2
 m_3
 m_4
 m_5
 m_6
 m_7
 m_8
 $x_{1_1}(1.5)$
 $x_{1_2}(1.5)$
 $x_{1_3}(1.5)$
 $x_{1_4}(1.5)$
 $x_{1_5}(1.5)$
 $x_{1_6}(1.5)$
 $x_{1_7}(1.5)$
 $x_{1_8}(1.5)$
 $x_{2_1}(1.5)$
 $x_{2_2}(1.5)$
 $x_{2_3}(1.5)$
 $x_{2_4}(1.5)$
 $x_{2_5}(1.5)$
 $x_{2_6}(1.5)$
 $x_{2_7}(1.5)$
 $x_{2_8}(1.5)$
 $x_{3_1}(1.5)$
 $x_{3_2}(1.5)$
 $x_{3_3}(1.5)$
 $x_{3_4}(1.5)$
 $x_{3_5}(1.5)$
 $x_{3_6}(1.5)$
 $x_{3_7}(1.5)$
 $x_{3_8}(1.5)$
 $x_{4_1}(1.5)$
 $x_{4_2}(1.5)$
 $x_{4_3}(1.5)$
 $x_{4_4}(1.5)$
 $x_{4_5}(1.5)$
 $x_{4_6}(1.5)$
 $x_{4_7}(1.5)$
 $x_{4_8}(1.5)$
 $x_{5_1}(0.75)$
 $x_{5_2}(0.75)$
 $x_{5_3}(0.75)$

x_5_4(0.75)
x_5_5(0.75)
x_5_6(0.75)
x_5_7(0.75)
x_5_8(0.75)
x_6_1(0.75)
x_6_2(0.75)
x_6_3(0.75)
x_6_4(0.75)
x_6_5(0.75)
x_6_6(0.75)
x_6_7(0.75)
x_6_8(0.75)
x_7_1(0.75)
x_7_2(0.75)
x_7_3(0.75)
x_7_4(0.75)
x_7_5(0.75)
x_7_6(0.75)
x_7_7(0.75)
x_7_8(0.75)
x_8_1(0.75)
x_8_2(0.75)
x_8_3(0.75)
x_8_4(0.75)
x_8_5(0.75)
x_8_6(0.75)
x_8_7(0.75)
x_8_8(0.75)
x_9_1(0.75)
x_9_2(0.75)
x_9_3(0.75)
x_9_4(0.75)
x_9_5(0.75)
x_9_6(0.75)
x_9_7(0.75)
x_9_8(0.75)
x_10_1(0.75)
x_10_2(0.75)
x_10_3(0.75)
x_10_4(0.75)
x_10_5(0.75)
x_10_6(0.75)

$x_{10_7}(0.75)$
 $x_{10_8}(0.75)$
 $x_{11_1}(0.22)$
 $x_{11_2}(0.22)$
 $x_{11_3}(0.22)$
 $x_{11_4}(0.22)$
 $x_{11_5}(0.22)$
 $x_{11_6}(0.22)$
 $x_{11_7}(0.22)$
 $x_{11_8}(0.22)$
 $x_{12_1}(0.22)$
 $x_{12_2}(0.22)$
 $x_{12_3}(0.22)$
 $x_{12_4}(0.22)$
 $x_{12_5}(0.22)$
 $x_{12_6}(0.22)$
 $x_{12_7}(0.22)$
 $x_{12_8}(0.22)$
 $x_{13_1}(0.22)$
 $x_{13_2}(0.22)$
 $x_{13_3}(0.22)$
 $x_{13_4}(0.22)$
 $x_{13_5}(0.22)$
 $x_{13_6}(0.22)$
 $x_{13_7}(0.22)$
 $x_{13_8}(0.22)$
 $x_{14_1}(0.22)$
 $x_{14_2}(0.22)$
 $x_{14_3}(0.22)$
 $x_{14_4}(0.22)$
 $x_{14_5}(0.22)$
 $x_{14_6}(0.22)$
 $x_{14_7}(0.22)$
 $x_{14_8}(0.22)$
 $x_{15_1}(0.22)$
 $x_{15_2}(0.22)$
 $x_{15_3}(0.22)$
 $x_{15_4}(0.22)$
 $x_{15_5}(0.22)$
 $x_{15_6}(0.22)$
 $x_{15_7}(0.22)$
 $x_{15_8}(0.22)$
 $x_{16_1}(0.22)$

x_16_2(0.22)
x_16_3(0.22)
x_16_4(0.22)
x_16_5(0.22)
x_16_6(0.22)
x_16_7(0.22)
x_16_8(0.22)
x_17_1(0.22)
x_17_2(0.22)
x_17_3(0.22)
x_17_4(0.22)
x_17_5(0.22)
x_17_6(0.22)
x_17_7(0.22)
x_17_8(0.22)
x_18_1(0.22)
x_18_2(0.22)
x_18_3(0.22)
x_18_4(0.22)
x_18_5(0.22)
x_18_6(0.22)
x_18_7(0.22)
x_18_8(0.22)
x_19_1(0.22)
x_19_2(0.22)
x_19_3(0.22)
x_19_4(0.22)
x_19_5(0.22)
x_19_6(0.22)
x_19_7(0.22)
x_19_8(0.22)
x_20_1(0.22)
x_20_2(0.22)
x_20_3(0.22)
x_20_4(0.22)
x_20_5(0.22)
x_20_6(0.22)
x_20_7(0.22)
x_20_8(0.22)
x_21_1(0.22)
x_21_2(0.22)
x_21_3(0.22)
x_21_4(0.22)

```

x_21_5(0.22)
x_21_6(0.22)
x_21_7(0.22)
x_21_8(0.22)
x_22_1(0.22)
x_22_2(0.22)
x_22_3(0.22)
x_22_4(0.22)
x_22_5(0.22)
x_22_6(0.22)
x_22_7(0.22)
x_22_8(0.22)
End

```

5 Explication du fichier CPLEX LP généré

Le fichier généré est divisé en trois sections : *Minimize*, *Subject To* et *Binary*. Dans la section *Minimize* nous trouvons la fonction objectif, dans ce projet cette fonction minimisera le nombre de planches à utiliser. *Subject To* est une section dans laquelle nos contraintes seront regroupées, comme il n'y a pas de mots clés pour représenter une somme nous le ferons explicitement, ce qui explique les $c1_j$ et $c2_i$ où j est une planche appartenant à M et i un panneau appartenant à N . Enfin, la section *Binary* contient toutes les variables binaires utilisées dans notre modélisation, la nomenclature de ces variables a été choisie comme suit :

m_j : représente la planche j

$x_{i_j}(l_i)$: représente le panneau i de longueur l_i découpé de la planche j

6 Log de GLPK obtenu

```

Problem:
Rows:      30
Columns:    184 (184 integer, 184 binary)
Non-zeros:  360
Status:     INTEGER OPTIMAL
Objective:  obj = 4 (MINimum)

```

No.	Row name	Activity	Lower bound	Upper bound
1	c1_1	0	0	
2	c1_2	0.21	0	
3	c1_3	1.75	0	
4	c1_4	0	0	
5	c1_5	0	0	
6	c1_6	0	0	
7	c1_7	0.78	0	
8	c1_8	0.12	0	
9	c2_1	1	1	=
10	c2_2	1	1	=
11	c2_3	1	1	=
12	c2_4	1	1	=
13	c2_5	1	1	=
14	c2_6	1	1	=
15	c2_7	1	1	=
16	c2_8	1	1	=
17	c2_9	1	1	=
18	c2_10	1	1	=
19	c2_11	1	1	=
20	c2_12	1	1	=
21	c2_13	1	1	=
22	c2_14	1	1	=
23	c2_15	1	1	=
24	c2_16	1	1	=
25	c2_17	1	1	=
26	c2_18	1	1	=
27	c2_19	1	1	=
28	c2_20	1	1	=
29	c2_21	1	1	=
30	c2_22	1	1	=

No.	Column name	Activity	Lower bound	Upper bound
-----	-------------	----------	-------------	-------------

1	m_1	*	0	0	1
2	m_2	*	1	0	1
3	m_3	*	1	0	1
4	m_4	*	0	0	1
5	m_5	*	0	0	1
6	m_6	*	0	0	1
7	m_7	*	1	0	1
8	m_8	*	1	0	1
9	x_1_1(1.5)	*	0	0	1
10	x_2_1(1.5)	*	0	0	1
11	x_3_1(1.5)	*	0	0	1
12	x_4_1(1.5)	*	0	0	1
13	x_5_1(0.75)	*	0	0	1
14	x_6_1(0.75)	*	0	0	1
15	x_7_1(0.75)	*	0	0	1
16	x_8_1(0.75)	*	0	0	1
17	x_9_1(0.75)	*	0	0	1
18	x_10_1(0.75)	*	0	0	1
19	x_11_1(0.22)	*	0	0	1
20	x_12_1(0.22)	*	0	0	1
21	x_13_1(0.22)	*	0	0	1
22	x_14_1(0.22)	*	0	0	1
23	x_15_1(0.22)	*	0	0	1
24	x_16_1(0.22)	*	0	0	1
25	x_17_1(0.22)	*	0	0	1
26	x_18_1(0.22)	*	0	0	1
27	x_19_1(0.22)	*	0	0	1
28	x_20_1(0.22)	*	0	0	1
29	x_21_1(0.22)	*	0	0	1
30	x_22_1(0.22)	*	0	0	1
31	x_1_2(1.5)	*	0	0	1
32	x_2_2(1.5)	*	0	0	1
33	x_3_2(1.5)	*	0	0	1

34	x_4_2(1.5)	*	0	0	1
35	x_5_2(0.75)	*	0	0	1
36	x_6_2(0.75)	*	1	0	1
37	x_7_2(0.75)	*	1	0	1
38	x_8_2(0.75)	*	0	0	1
39	x_9_2(0.75)	*	0	0	1
40	x_10_2(0.75)	*	1	0	1
41	x_11_2(0.22)	*	1	0	1
42	x_12_2(0.22)	*	1	0	1
43	x_13_2(0.22)	*	1	0	1
44	x_14_2(0.22)	*	1	0	1
45	x_15_2(0.22)	*	1	0	1
46	x_16_2(0.22)	*	1	0	1
47	x_17_2(0.22)	*	1	0	1
48	x_18_2(0.22)	*	0	0	1
49	x_19_2(0.22)	*	0	0	1
50	x_20_2(0.22)	*	0	0	1
51	x_21_2(0.22)	*	0	0	1
52	x_22_2(0.22)	*	0	0	1
53	x_1_3(1.5)	*	0	0	1
54	x_2_3(1.5)	*	0	0	1
55	x_3_3(1.5)	*	0	0	1
56	x_4_3(1.5)	*	0	0	1
57	x_5_3(0.75)	*	1	0	1
58	x_6_3(0.75)	*	0	0	1
59	x_7_3(0.75)	*	0	0	1
60	x_8_3(0.75)	*	1	0	1
61	x_9_3(0.75)	*	1	0	1
62	x_10_3(0.75)	*	0	0	1
63	x_11_3(0.22)	*	0	0	1
64	x_12_3(0.22)	*	0	0	1
65	x_13_3(0.22)	*	0	0	1
66	x_14_3(0.22)	*	0	0	1
67	x_15_3(0.22)	*	0	0	1

68	x_16_3(0.22)	*	0	0	1
69	x_17_3(0.22)	*	0	0	1
70	x_18_3(0.22)	*	0	0	1
71	x_19_3(0.22)	*	0	0	1
72	x_20_3(0.22)	*	0	0	1
73	x_21_3(0.22)	*	0	0	1
74	x_22_3(0.22)	*	0	0	1
75	x_1_4(1.5)	*	0	0	1
76	x_2_4(1.5)	*	0	0	1
77	x_3_4(1.5)	*	0	0	1
78	x_4_4(1.5)	*	0	0	1
79	x_5_4(0.75)	*	0	0	1
80	x_6_4(0.75)	*	0	0	1
81	x_7_4(0.75)	*	0	0	1
82	x_8_4(0.75)	*	0	0	1
83	x_9_4(0.75)	*	0	0	1
84	x_10_4(0.75)	*	0	0	1
85	x_11_4(0.22)	*	0	0	1
86	x_12_4(0.22)	*	0	0	1
87	x_13_4(0.22)	*	0	0	1
88	x_14_4(0.22)	*	0	0	1
89	x_15_4(0.22)	*	0	0	1
90	x_16_4(0.22)	*	0	0	1
91	x_17_4(0.22)	*	0	0	1
92	x_18_4(0.22)	*	0	0	1
93	x_19_4(0.22)	*	0	0	1
94	x_20_4(0.22)	*	0	0	1
95	x_21_4(0.22)	*	0	0	1
96	x_22_4(0.22)	*	0	0	1
97	x_1_5(1.5)	*	0	0	1
98	x_2_5(1.5)	*	0	0	1
99	x_3_5(1.5)	*	0	0	1
100	x_4_5(1.5)	*	0	0	1
101	x_5_5(0.75)	*	0	0	1

102	x_6_5(0.75)	*	0	0	1
103	x_7_5(0.75)	*	0	0	1
104	x_8_5(0.75)	*	0	0	1
105	x_9_5(0.75)	*	0	0	1
106	x_10_5(0.75)	*	0	0	1
107	x_11_5(0.22)	*	0	0	1
108	x_12_5(0.22)	*	0	0	1
109	x_13_5(0.22)	*	0	0	1
110	x_14_5(0.22)	*	0	0	1
111	x_15_5(0.22)	*	0	0	1
112	x_16_5(0.22)	*	0	0	1
113	x_17_5(0.22)	*	0	0	1
114	x_18_5(0.22)	*	0	0	1
115	x_19_5(0.22)	*	0	0	1
116	x_20_5(0.22)	*	0	0	1
117	x_21_5(0.22)	*	0	0	1
118	x_22_5(0.22)	*	0	0	1
119	x_1_6(1.5)	*	0	0	1
120	x_2_6(1.5)	*	0	0	1
121	x_3_6(1.5)	*	0	0	1
122	x_4_6(1.5)	*	0	0	1
123	x_5_6(0.75)	*	0	0	1
124	x_6_6(0.75)	*	0	0	1
125	x_7_6(0.75)	*	0	0	1
126	x_8_6(0.75)	*	0	0	1
127	x_9_6(0.75)	*	0	0	1
128	x_10_6(0.75)	*	0	0	1
129	x_11_6(0.22)	*	0	0	1
130	x_12_6(0.22)	*	0	0	1
131	x_13_6(0.22)	*	0	0	1
132	x_14_6(0.22)	*	0	0	1
133	x_15_6(0.22)	*	0	0	1
134	x_16_6(0.22)	*	0	0	1
135	x_17_6(0.22)	*	0	0	1

136	x_18_6(0.22)	*	0	0	1
137	x_19_6(0.22)	*	0	0	1
138	x_20_6(0.22)	*	0	0	1
139	x_21_6(0.22)	*	0	0	1
140	x_22_6(0.22)	*	0	0	1
141	x_1_7(1.5)	*	0	0	1
142	x_2_7(1.5)	*	0	0	1
143	x_3_7(1.5)	*	1	0	1
144	x_4_7(1.5)	*	1	0	1
145	x_5_7(0.75)	*	0	0	1
146	x_6_7(0.75)	*	0	0	1
147	x_7_7(0.75)	*	0	0	1
148	x_8_7(0.75)	*	0	0	1
149	x_9_7(0.75)	*	0	0	1
150	x_10_7(0.75)	*	0	0	1
151	x_11_7(0.22)	*	0	0	1
152	x_12_7(0.22)	*	0	0	1
153	x_13_7(0.22)	*	0	0	1
154	x_14_7(0.22)	*	0	0	1
155	x_15_7(0.22)	*	0	0	1
156	x_16_7(0.22)	*	0	0	1
157	x_17_7(0.22)	*	0	0	1
158	x_18_7(0.22)	*	0	0	1
159	x_19_7(0.22)	*	0	0	1
160	x_20_7(0.22)	*	0	0	1
161	x_21_7(0.22)	*	0	0	1
162	x_22_7(0.22)	*	1	0	1
163	x_1_8(1.5)	*	1	0	1
164	x_2_8(1.5)	*	1	0	1
165	x_3_8(1.5)	*	0	0	1
166	x_4_8(1.5)	*	0	0	1
167	x_5_8(0.75)	*	0	0	1
168	x_6_8(0.75)	*	0	0	1
169	x_7_8(0.75)	*	0	0	1

170	x_8_8(0.75)	*	0	0	1
171	x_9_8(0.75)	*	0	0	1
172	x_10_8(0.75)	*	0	0	1
173	x_11_8(0.22)	*	0	0	1
174	x_12_8(0.22)	*	0	0	1
175	x_13_8(0.22)	*	0	0	1
176	x_14_8(0.22)	*	0	0	1
177	x_15_8(0.22)	*	0	0	1
178	x_16_8(0.22)	*	0	0	1
179	x_17_8(0.22)	*	0	0	1
180	x_18_8(0.22)	*	1	0	1
181	x_19_8(0.22)	*	1	0	1
182	x_20_8(0.22)	*	1	0	1
183	x_21_8(0.22)	*	1	0	1
184	x_22_8(0.22)	*	0	0	1

Integer feasibility conditions:

KKT.PE: max.abs.err = 8.88e-16 on row 2
max.rel.err = 9.87e-17 on row 2
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

7 Explication du retour obtenu dans le fichier log

7.1 Problème

Pour un problème avec 8 planches de 4 mètres à partir desquelles 4 panneaux de 1,5 mètres, 6 panneaux de 75 centimètres et 12 panneaux de 22 centimètres doivent être dé-

coupés, ce fichier log donne des informations générales sur ce problème et sa résolution.

7.2 Fonction objectif I

Objective: obj = 4 (MINimum)

La valeur de la fonction objectif indique que quatre est le nombre minimum de planches qui devront être utilisées. Par la suite, une solution détaillée détermine un plan de découpe des panneaux pour le nombre de planches utilisées.

No.	Column name	Activity	Lower bound	Upper bound
1	m_1	*	0	1
2	m_2	*	1	1
3	m_3	*	1	1
4	m_4	*	0	1
5	m_5	*	0	1
6	m_6	*	0	1
7	m_7	*	1	1
8	m_8	*	1	1

Comme indiqué dans la colonne *Activity*, la solution trouvée utilise les planches 2,3,7 et 8 où la valeur est égale à 1, ce qui est cohérent avec la valeur de la fonction objectif.

7.3 Contrainte 1

Prenons la planche 2 comme exemple, nous voulons obtenir son plan de découpe, nous allons donc extraire ces lignes ci-dessous du fichier log.

No.	Row name	Activity	Lower bound	Upper bound
2	c1_2	0.21	0	
No.	Column name	Activity	Lower bound	Upper bound
36	x_6_2(0.75)	*	1	1
37	x_7_2(0.75)	*	1	1
40	x_10_2(0.75)	*	1	1

41	x_11_2(0.22) *	1	0	1
42	x_12_2(0.22) *	1	0	1
43	x_13_2(0.22) *	1	0	1
44	x_14_2(0.22) *	1	0	1
45	x_15_2(0.22) *	1	0	1
46	x_16_2(0.22) *	1	0	1
47	x_17_2(0.22) *	1	0	1

La première ligne a comme nom de variable $x_{6_2}(0.75)$ ce qui signifie selon la nomenclature établie au début que le panneau 6 d'une longueur de 0.75 mètre a été découpé de la planche 2. Suivant le même principe, nous déduirons que la planche 2 a été utilisée pour découper 3 panneaux de 0,75 mètre et 7 panneaux de 0,22 mètre. La contrainte 1 pour la planche 2 ($c1_2$) nous donne une *Activity* de 0,21 qui correspond à la soustraction de la longueur de la planche de la somme des panneaux découpés, soit $4 - 3 \cdot 0,75 - 7 \cdot 0,22 = 0,21$.

7.4 Contrainte 2

Prenons un panneau de ceux utilisés dans la découpe de la planche 2, dans notre cas nous serons intéressés par le panneau 6, pour mieux visualiser la résolution du fichier log nous prenons ces morceaux de lignes.

No.	Row name	Activity	Lower bound	Upper bound
14	c2_6	1	1	=
No.	Column name	Activity	Lower bound	Upper bound
14	x_6_1(0.75) *	0	0	1
36	x_6_2(0.75) *	1	0	1
58	x_6_3(0.75) *	0	0	1
80	x_6_4(0.75) *	0	0	1
102	x_6_5(0.75) *	0	0	1
124	x_6_6(0.75) *	0	0	1
146	x_6_7(0.75) *	0	0	1
168	x_6_8(0.75) *	0	0	1

Comme le montre la colonne *Activity* pour $c2_6$ le panneau a bien été découpé et seulement une fois, pour vérifier que la contrainte a été respectée nous passons en revue par les lignes restantes. En examinant l'activité du panneau 6, nous voyons qu'il n'est découpé qu'une seule fois et qu'il l'a été depuis la planche 2 ($x_{6_2}(0.75) = 1$).

7.5 Plan de découpe

En répétant la même analyse sur les autres planches et panneaux, nous obtenons le plan de découpe suivant :

<i>Planche</i>	1	2	3	4	5	6	7	8
<i>Panneau de 1.5</i>	0	0	0	0	0	0	2	2
<i>Panneau de 0.75</i>	0	3	3	0	0	0	0	0
<i>Panneau de 0.22</i>	0	7	0	0	0	0	1	4