TP : Problème de découpe optimale (2 séances)

Certains matériaux (papiers, tissus, feuilles métalliques) sont fabriqués sous la forme de larges rouleaux et ensuite débités en rouleaux moins larges. Typiquement, on a un petit nombre de largeurs *initiales* standards fixées par le fabriquant, un grand nombre de largeurs *finales* choisies par les clients. Par exemple, un rouleau initial de 3 m. peut être débité en 2 rouleaux de 80 cm., 1 de 120 cm et il reste une chute de 20 cm.

Etant donné un ensemble de commandes, le problème qui nous intéresse consiste à choisir les découpes de façon à minimiser le nombre de rouleaux initiaux à découper.
Par exemple, supposons que la largeur initiale des rouleaux est unique : 100 cm. Les commandes sont résumées dans le tableau suivant :

97	rouleaux de largeur	45 cm.
610	rouleaux de largeur	36 cm.
395	rouleaux de largeur	31 cm.
211	rouleaux de largeur	14 cm.

On a commencé par rassembler les 37 façons de découper un rouleau initial dans la table ci-dessous

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a_{1j}	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
a_{2j}	0	1	1	0	0	0	0	0	0	2	2	2	1	1	1	1	1	1	1	1
a_{3j}	0	0	0	1	1	0	0	0	0	0	0	0	2	1	1	1	0	0	0	0
a_{4j}	0	1	0	1	0	3	2	1	0	2	1	0	0	2	1	0	4	3	2	1

j	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
a_{1j}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a_{2j}	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a_{3j}	0	3	2	2	2	1	1	1	1	1	0	0	0	0	0	0	0
a_{4j}	0	0	2	1	0	4	3	2	1	0	7	6	5	4	3	2	1

Il reste à savoir combien de fois, on utilise chacun de ces motifs. Les variables de décision du problèmes sont les x_j pour $j = 1, 2, \ldots, 37$ qui représentent le nombre de fois où on utilise le motif j. Pour satisfaire toutes les demandes, on doit avoir :

$$\sum_{j=1}^{37} a_{ij} x_j = b_i \quad (i = 1, 2, 3, 4) \tag{1}$$

avec $b_1 = 97$, $b_2 = 610$, $b_3 = 395$ et $b_4 = 211$. Nous sommes donc conduit à résoudre le programme linéaire (P) :

Minimiser $\sum_{j=1}^{37} x_j$ *sous les contraintes* (1) *et* $x_j \geq 0 \quad (j = 1, 2, \ldots, n)$.

La solution de ce programme linéaire n'est pas forcément entière. Cependant, pour ce problème particulier, la solution optimale fractionnaire est souvent proche de la solution optimale. Par exemple, ici, la solution optimale du programme linéaire est

$$x_1^* = 48,5 \;\; x_{10}^* = 105,75 \;\; x_{12}^* = 100,75 \;\; x_{13}^* = 197,5.$$

En approximation cette solution par défaut, on obtient une solution qui consiste à découper 450 rouleaux initiaux et qui couvre presque toutes les commandes. Les demandes insatisfaites peuvent être couvertes avec seulement trois rouleaux supplémentaires. Ce qui conduit a une solution avec 453 rouleaux. Le coût de la solution optimale du programme linéaire est 452.25. Par conséquent, la solution avec 453 finaux est optimale. Ce n'est pas le cas en général, mais ce type d'approximation conduit le plus souvent en pratique à une solution satisfaisante.

Une difficulté persiste lorsque l'on s'intéresse à des problèmes de tailles réelles, le nombre de variables devient astronomique (jusqu'à 100 millions de variables). C'est ce qui a conduit Gilmore et Gomory à introduire la technique de génération de colonnes. L'idée est de considérer seulement un sous-ensemble de motifs à la fois et on en génère d'autres au fur et à mesure uniquement si c'est nécessaire.
Le but de ce tp est d'écrire un programme implémentant cet algorithme, en utilisant la librairie C de programmation mathématique GLPK. Vous pouvez coder avec le langage de programmation de votre choix, du moment qu'il permet d'utiliser cette librairie (C, java, Haskell,...).
Vouz pouvez consulter cet exemple du wikibook sur GLPK pour comprendre comment la librairie s'utilise.
`http://en.wikibooks.org/wiki/GLPK/Using_the_GLPK_callable_library`
La principale difficulté est de comprendre comment la matrice du programme linéaire est représentée. On utilise 3 tableaux, l'un contenant les coefficients apparaissant dans la matrice `coef`, le deuxième `row` donne la ligne et le troisième `column` donne la colonne du coefficient de même indice. Ainsi, $M_{c,r}$ est `coef[i]` lorsque `column[i]` $= c$ et `row[i]` $= r$.

Génération de colonnes

On considère un problème de découpe où r est la largeur des rouleaux initiaux et b_i rouleaux finaux de largeur w_i sont commandés, pour $i = 1, 2, \ldots, m$. On a vu qu'on doit alors résoudre le programme linéaire suivant

Minimiser cx *sous les contraintes* $Ax = b,\ x \geq 0$.

où b un vecteur colonne dont les composantes sont $b_1, b_2, \ldots, b_m$; c un vecteur ligne dont les composantes sont $1, 1, \ldots, 1$; chaque colonne $a = [a_1, a_2, \ldots, a_m]$ de la matrice A décrit un motif. Par conséquent, a est une colonne de A si et seulement si $\sum_{i=1}^{m} w_i a_i \leq r$.

Dans la méthode simplex, à chaque itération on cherche une colonne hors base dont le coût réduit est négatif. (Rappel : le coût réduit d'une colonne est donnée par $\bar{c}_j = c_j - \sum_{i=1}^{m} y_i a_i$) où a est la colonne en question et $(y_1, \ldots, y_m)$ est la solution optimale du dual.
Ici, le coût réduit de la colonne $[a_1, a_2, \ldots, a_m]$ est $1 - \sum_{i=1}^{m} y_i a_i$. Donc, on cherche des entiers positifs $a_1, a_2, \ldots, a_m$ tels que $\sum_{i=1}^{m} w_i a_i \leq r$ et $\sum_{i=1}^{m} y_i a_i > 1$. Pour trouver, un tel ensemble d'entiers, on peut résoudre le programme linéaire en nombres entiers suivant (problème de sac à dos) :

Maximiser $\sum_{i=1}^{m} y_i a_i$ *sous les contraintes* $\sum_{i=1}^{m} w_i a_i \leq r$ *et* $a_i \geq 0$ *entier* $(i = 1, 2, \ldots m)$.

Si la valeur de la solution optimale de ce problème est supérieure 1, le motif correspondant doit être ajouté au programme linéaire sinon la solution courante est optimale.
Le fonctionnement de l'algorithme est le suivant :

1. On sélectionne un sous-ensemble initial de colonnes.
2. On résout le programme linéaire réduit correspondant.
3. On récupère la solution duale optimale du programme linéaire réduit.
4. On résout le problème de sac à dos pour lequel le poids de chacun des objets est défini par la valeur de la variable duale correspondante.
5. Si la solution optimale du problème de sac à dos a un coût supérieur à 1, on ajoute la colonne correspondante au programme linéaire et on retourne en 2. Sinon la solution courante du programme linéaire réduit est optimale pour le programme linéaire complet. STOP.