MPRO - Mise à niveau

EXERCICES - Programmation linéaire en nombres entiers

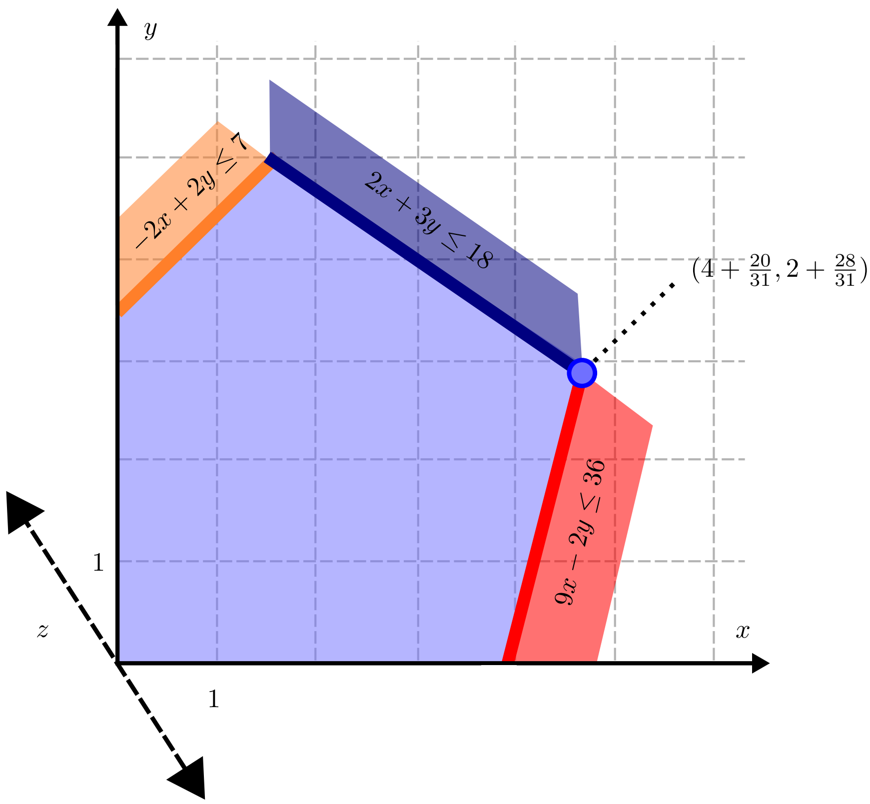
Exercice 1 : Branch & Bound et PLNE

La méthode de Dakin est un algorithme de branch-and-bound dans lequel le branchement est effectué sur la variable qui a la plus grande partie fractionnaire dans la solution en cours.
Utiliser cette méthode pour trouver la solution optimale de l'exemple suivant :

$$
\begin{array}{lrrcr}
\max z = & 3x & +2y & & \\
\text{s.c.} & -2x & +2y & \leq & 7 \\
 & 2x & +3y & \leq & 18 \\
 & 9x & -2y & \leq & 36 \\
 & x, & y & \in & \mathbb{N}
\end{array}
$$

Les solutions des programmes linéaires continus à résoudre en chaque nœud seront trouvées directement de façon géométrique sur la figure suivante. La solution continue du problème à la racine est obtenue au point indiqué.
Indication : Voici une liste de coordonnées à considérer lors de la résolution (pas nécessairement dans cet ordre) :

- $(4; 3+\frac{1}{3})$ $z = 18 + \frac{2}{3}$
- $(4+\frac{4}{9}; 2)$ $z = 17 + \frac{1}{3}$
- $(4+\frac{1}{2}; 3)$ $z = 19 + \frac{1}{2}$
- $(3; 4)$ $z = 17$



Exercice 2 : Branch & Bound et problème du sac à dos

Pour embarquer dans l'avion, votre valise ne doit pas peser plus de K kg. Il ne va pas être possible d'emporter les n objets que vous vouliez y placer et vous devez faire des choix. Vous avez pesé chaque

objet et vous notez p_i le poids de l'objet $i \in \{1, ..., n\}$. Pour optimiser le contenu de la valise, vous avez attribué à chaque objet i une note d'utilité c_i entre 1 et 20. Vous voulez maximiser l'utilité globale de la valise, égale à la somme des utilités des objets emportés.
Question 1 Pour chaque objet i, on définit la variable x_i égale à 1 si l'objet i est mis dans la valise et 0 sinon. Ecrire le programme mathématique P à résoudre.
Question 2 Ecrire le programme obtenu si $K = 17$ et que vous avez 4 objets de poids respectifs 3, 7, 9 et 6 kg et d'utilités respectives 8, 18, 20 et 11.
Question 3 On suppose maintenant que les objets sont fractionnables. Soit P_R le programme P dans lequel on a remplacé $x_i \in \{0, 1\}$ par $x_i \in [0, 1]$, pour tout $i \in \{1, ..., n\}$.
Le programme P_R (relaxation continue de P) peut se résoudre de la façon suivante :

1. Trier les ratios $\frac{c_i}{p_i}$ par ordre décroissant.
2. Mettre les objets dans le sac dans cet ordre jusqu'à ce qu'on dépasse K.
3. Mettre une fraction du premier objet qui dépasse K dans le sac de façon à compléter le poids de la valise (jusqu'à K).

Soit $imax$ l'indice du dernier objet mis complètement dans le sac (*i.e.*, $\sum_{i=1}^{imax} p_i \leq K$ et $\sum_{i=1}^{imax+1} p_i > K$). On a donc :

— les objets 1 à $imax$ sont entièrement dans le sac ($x_i = 1$ pour $i \in \{1, ..., imax\}$) ;
— l'objet $imax + 1$ est partiellement dans le sac ($x_{imax+1} = \frac{K - \sum_{i=1}^{imax} p_i}{p_{imax+1}}$) ;
— les autres objets ne sont pas dans le sac ($x_i = 0$ pour $i \in \{imax + 2, ..., n\}$).

Quelle est la solution de P_R pour l'instance mentionnée dans la question précédente ?
Question 4 Résoudre le problème P pour cette instance par une méthode arborescente. En chaque nœud on obtient un majorant en résolvant la relaxation linéaire du problème en ce nœud.
Indications : la séparation (en 2 sous-arbres) se fait sur la variable x_i non entière. On déduit éventuellement des implications consécutives à chaque choix (c'est-à-dire qu'on fixe à 0 les variables associées à des objets qui ne pourront plus entrer dans la valise une fois que ce choix est fait).