Master TET, ENPC
Recherche Opérationnelle

2025-26

TD2 : Optimisation linéaire réelle et en nombres entiers

Exercice 1. *Simplexe pour un PL*

On considère le programme linéaire (P) suivant :

$$\max_{(x_1,x_2)\in\mathcal{D}} F(\mathbf{x}) = 2x_1 + x_2 \quad \text{avec} \quad \mathcal{D} = \{(x_1,x_2)\in\mathbb{R}^2, x_1 \geq 0, x_2 \geq 0 \text{ et } x_1 \leq 1,\ -x_1 + x_2 \leq 1\}$$

1. Résoudre graphiquement (P).
2. Résoudre (P) par la méthode des dictionnaires de l'algorithme du simplexe.

Exercice 2. *Application du Théorème des écarts complémentaires*

On considère le programme linéaire (P) suivant

$$\max F(\mathbf{x}) = 4x_1 + 5x_2$$
$$\begin{cases} 3x_1 + x_2 \leq 1 \\ x_1 + 4x_2 \leq 1 \\ x_1,\ x_2 \geq 0 \end{cases}$$

A partir du théorème des écarts complémentaires, montrer que $\mathbf{x} = (\frac{3}{11}, \frac{2}{11})$ est solution optimale de (P).

Exercice 3. *Programmation linéaire en variables* $\{0,1\}$ *– Problème de sac à dos.*

Soit le problème de "sac-à-dos" suivant :

$$\max\left[F(\mathbf{x}) = 6x_1 + 9x_2 + 8x_3 + 4x_4\right]$$
$$\begin{cases} 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 8 \\ 3x_1 + 6x_2 + 4x_3 + 3x_4 \leq 7 \\ x_1, x_2, x_3, x_4 \in \{0,1\} \end{cases}$$

Résoudre ce problème par une procédure de Branch-and-Bound.

Exercice 4. *Méthode des coupes entières.*

On considère le programme linéaire en nombres entiers suivant

$$(P) \quad \begin{cases} \max\limits_{\mathbf{x}=(x_1,x_2)} \left[F(x) = x_1 + 10x_2\right] \\ A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}, \\ x_1,\ x_2 \text{ entiers} \end{cases} \qquad \text{avec } A = \begin{pmatrix} -2 & 1 \\ 2 & 18 \\ 6 & 1 \end{pmatrix} \text{ et } \mathbf{b} = \begin{pmatrix} 2 \\ 93 \\ 78 \end{pmatrix}.$$

Utiliser une procédure de séparation avec des coupes entières (Branch-and-Cut) pour résoudre (P).

Rappel de la méthode :

1. On commence par résoudre le problème (P_0) qui est le problème (P) relaxé i.e. sans la contrainte "x_1 et x_2 entiers".
2. On examine si la solution optimale (x_1^*, x_2^*) de (P_0) est entière.
   (a) Si c'est le cas alors on arrête.

(b) Sinon, on considère la *première* variable optimale non entière x_i^* ($i = 1$ ou 2) et on construit deux autres problèmes (P_1) et (P_2) en rajoutant à (P_0) respectivement les contraintes

$$x_i \leq \lfloor x_i^* \rfloor \text{ pour } (P_1), \quad x_i \geq \lfloor x_i^* \rfloor + 1 \text{ pour } (P_2) \tag{1}$$

où $\lfloor \cdot \rfloor$ désigne la partie entière inférieure.

(c) On résout les problèmes *auxiliaires* (P_1) et (P_2).

(d) On sélectionne le problème (P_1) ou (P_2) réalisable qui possède le coût F le plus grand et on retourne en 2) avec ce problème à la place de (P_0).

On associe à cette méthode une procédure de type Branch-and-Bound. La phase de séparation (branch) correspond à la construction des deux problèmes (P_1) et (P_2). La phase d'évaluation (bound) correspond à la détermination d'une valeur $\overline{F}$ du coût pour une solution réalisable **entière** particulière. Initialement, on prend $\overline{F} = +\infty$ et on actualise $\overline{F}$ quand une solution optimale *entière* d'un problème auxiliaire est obtenue. La valeur $\overline{F}$ permet de déterminer si on arrête d'examiner un sommet de l'arbre.

Vous utiliserez les indications ci-dessous qui fournissent les solutions optimales des programmes linéaires (sans contrainte d'intégrité sur les variables) de la forme

$$(PL) \quad \begin{cases} \max\limits_{\mathbf{x}} \,[F(\mathbf{x}) = x_1 + 10x_2] \\ \quad A\mathbf{x} \leq \mathbf{b}, \\ \quad C\mathbf{x} \leq \mathbf{d}, \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

où des contraintes supplémentaires (les coupes entières) ont été rajoutées avec la matrice C et le vecteur $\mathbf{d}$. Les indications sont données pour certaines valeurs de C et $\mathbf{d}$.

Indications :

1. La solution optimale de (P) relaxé (i.e. sans contrainte d'intégrité sur les variables) est donnée par $x_1^* = 3/2$, $x_2^* = 5$ et $F_{\max}^* = 51,5$.
2. Avec $C = \begin{pmatrix} 1 & 0 \end{pmatrix}$ et $\mathbf{d} = 1$, la solution optimale de (PL) est $x_1^* = 1$, $x_2^* = 4$ et $F_{\max}^* = 41$.
3. Avec $C = \begin{pmatrix} -1 & 0 \end{pmatrix}$ et $\mathbf{d} = -2$, la solution optimale de (PL) est $x_1^* = 2$, $x_2^* = 89/18 \simeq 4,944$ et $F_{\max}^* = 463/9 \simeq 51,444$.
4. Avec $C = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ 4 \end{pmatrix}$, la solution optimale de (PL) est $x_1^* = 10,5$, $x_2^* = 4$ et $F_{\max}^* = 50,5$.
5. Avec $C = \begin{pmatrix} -1 & 0 \\ -1 & 0 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ -5 \end{pmatrix}$, le (PL) n'a pas de solution réalisable.
6. Avec $C = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ 4 \\ 10 \end{pmatrix}$, la solution optimale de (PL) est $x_1^* = 10$, $x_2^* = 4$ et $F_{\max}^* = 50$.
7. Avec $C = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & 0 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ 4 \\ -11 \end{pmatrix}$, la solution optimale de (PL) est $x_1^* = 11$, $x_2^* \simeq 3,944$ et $F_{\max}^* = 50,444$.
8. Avec $C = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & 1 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ 4 \\ -11 \\ 3 \end{pmatrix}$, la solution optimale de (PL) est $x_1^* = 12,5$, $x_2^* = 3$ et $F_{\max}^* = 42,5$.
9. Avec $C = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$ et $\mathbf{d} = \begin{pmatrix} -2 \\ 4 \\ -11 \\ -4 \end{pmatrix}$, le (PL) n' a pas de solution réalisable.

Exercice 5. *Branch-&-Cut avec Python/Gurobi*

On considère le programme linéaire en nombres entiers suivant

$$(P)\ \begin{array}{l} \max_{(x_1,x_2)} (F(x_1,x_2) = 8x_1 + 11x_2) \\ \left\{ \begin{array}{l} x_1 + x_2 \leq 8 \\ 5x_1 + 8x_2 \leq 50 \\ x_1, x_2 \geq 0 \text{ entiers} \end{array} \right. \end{array}$$

1. Résoudre (P) par la méthode de Branch-&-Cut (cf. exercice précédent) avec des coupes entières, en utilisant Python/gurobipy.

   Pour résoudre les différents problèmes auxiliaires, utiliser le solveur Gurobi avec gurobipy dans Python, avec des variables *réelles*. Les coupes entières seront prises en compte comme des bornes (inférieures ou supérieures) sur les variables en définissant les vecteurs `lb` et `ub` [1] avec l'instance `.addMVar()` du modèle Gurobi :

   ```
   import gurobipy as gp
   m = gp.Model()
   x = m.addMVar(n, ub=[...])
   ```

   ou bien en utilisant la méthode `.setAttr()` sur les variables :

   ```
   x = m.addMVar(n)
   ub=[...]
   x.setAttr(gp.GRB.Attr.UB, ub)
   ```

2. Résoudre directement (P) avec gurobipy et des variables *entières*. Comparer avec la solution précédente.

Exercice 6. *Coupes de Gomory*

Soit le programme linéaire en nombres entiers suivant :

$$(P)\ \begin{array}{l} \max_{\mathbf{x}=(x_1,x_2)} [F(\mathbf{x}) = x_1 + 2x_2] \\ \left\{ \begin{array}{l} 7x_1 + 3x_2 \leq 21 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \ \textbf{entiers} \end{array} \right. \end{array}$$

On notera (P') le problème (P) relaxé i.e. sans la contrainte d'intégrité sur les variables.

1. Dessiner l'ensemble des solutions réalisables du problème (P'). Résoudre graphiquement les problèmes (P') et (P).
2. La solution optimale du problème (P') est $\mathbf{x}^* = (x_1^*, x_2^*)^\top = (12/7, 3)^\top \simeq (1.714, 3)^\top$. La composante x_1^* n'est pas entière. On cherche à déterminer une contrainte supplémentaire à ajouter à (P') pour forcer cette composante à devenir entière.
   (a) Ecrire le problème (P') sous forme standard en désignant par e_1 et e_2 les variables d'écarts et écrire le dernier dictionnaire du problème (P') ayant permis d'obtenir la solution optimale $\mathbf{x}^*$.
   (b) Ecrire la coupe de Gomory associée à la partie fractionnaire de x_1^* et montrer que cette coupe s'écrit
   $$e_1 + 4e_2 \geq 5 \tag{1}$$
   (c) Etablir que la contrainte précédente s'écrit aussi
   $$x_1 + x_2 \leq 4 \tag{2}$$
   (d) Ajouter la contrainte (2) au problème (P') pour obtenir un problème (P''). Résoudre graphiquement (P''). Commenter la solution optimale obtenue pour (P'').
3. *Coupe de Gomory et simplexe dual.* Ajouter au dernier dictionnaire associé à (P') la contrainte (1) avec une variable d'écart associée e_3 et réaliser une itération du simplexe dual. Retrouver la solution optimale *entière* déterminée graphiquement en 2.

---

1. S'il n'y a pas de borne supérieure sur une variable, vous pouvez quand même donner la valeur `Inf` correspondante dans le vecteur `ub`