
Génération de colonnes et lemme de Farkas

Hélène Toussaint, août 2014

But : montrer comment générer de nouvelles colonnes pour un programme linéaire non faisable en utilisant les multiplicateurs de Farkas.

Prérequis : bonne connaissance de la programmation linéaire

1 Rappel sur la génération de colonnes

Lorsque l'on a résoudre un programme linéaire (\mathcal{P}) comportant un ensemble de variables \mathcal{V} très (trop) grand on peut utiliser la génération de colonnes. Cette méthode consiste à construire un programme linéaire (PL) de taille réduite en utilisant seulement un sous-ensemble \mathcal{V}' de variables dans \mathcal{V} et à générer itérativement de nouvelles variables *intéressantes* appelées *variables améliorantes* (on parle souvent de *colonne* améliorante, une variable correspondant à une colonne de la matrice des contraintes).

Le PL comportant un nombre réduit de variables s'appelle *problème maître restreint* (\mathcal{PMR}). Le but de la génération de colonnes est de réussir à trouver la solution optimale du problème initial (\mathcal{P}) sans avoir à générer l'ensemble de toutes ses variables : il suffit en fait de générer (au moins) les variables formant une solution optimale. Pour trouver ces variables on résout le (\mathcal{PMR}) à l'optimal et on utilise la dualité pour générer une variable améliorante (ou même plusieurs). Ces nouvelles variables sont ajoutées au (\mathcal{PMR}), sa solution optimale est mise à jour (en même temps que les valeurs duales) et on recommence la recherche de variables améliorantes. On itère ainsi jusqu'à prouver qu'il n'y a plus de variable améliorante, la solution optimale du (\mathcal{PMR}) est alors la solution optimale de (\mathcal{P}).

Recherche de variables (ou colonnes) améliorantes pour un problème de minimisation

Soit 2 entiers non nuls m et n . On considère le (\mathcal{PMR}) général suivant dans lequel :

- $c \in \mathbb{R}^n$ (coefficients de la fonction objectif)
- $x \in \mathbb{R}^n$ (vecteur inconnu = variables)
- $A = (a_{ji}) \in \mathcal{M}_{mn}(\mathbb{R})$ (matrice des contraintes)
- $b \in \mathbb{R}^m$ (coefficients du second membre)
- chaque contrainte est associée à une variable duale $\lambda_j, j = 1, \dots, m$

$$\begin{aligned} &\text{Minimiser } \sum_{i=1}^n c_i x_i \\ &\text{sous } \forall j = 1, \dots, m, \sum_{i=1}^n a_{ji} x_i \geq b_j \\ &\quad \forall i = 1, \dots, n, x_i \geq 0 \end{aligned}$$

Remarque : bien sûr si on a des contraintes " \leq " il suffit de les multiplier par -1 pour se ramener à des contraintes " \geq ".

Une variable du PL est complètement définie par ses coefficients dans la matrice des contraintes. Par exemple la variable x_i est associée à la colonne $(a_{ji})_{j=1..m}$. La génération d'une nouvelle variable x_k implique donc le calcul des ses coefficients $(a_{jk})_{j=1..m}$ dans la matrice des contraintes.

Dans un problème de minimisation, on sait qu'une variable x_k est améliorante si son coût réduit \bar{c}_k est strictement négatif (voir [3] pour la démo et le calcul détaillé du coût réduit). On cherche donc un vecteur colonne $(a_{jk})_{j=1..m}$ telle que :

$$\begin{cases} \bar{c}_k = c_k - \sum_{j=1}^m \lambda_j a_{jk} < 0 \\ x_k \text{ est une variable pour le problème considéré} \end{cases}$$

Le problème de recherche d'une variable améliorante (couramment appelé "*Pricing problem*") revient donc à résoudre le système $c_k - \sum_{j=1}^m \lambda_j a_{jk} < 0$ dans lequel c_k (coefficient dans la fonction objectif de x_k) et $(\lambda_j)_{j=1..m}$ sont connus et $(a_{jk})_{j=1..m}$ (coefficients de la nouvelle colonne éventuelle de A) est l'inconnue. On considère souvent ce problème comme un problème de maximisation (PP) : $\text{Max } \sum_{j=1}^m \lambda_j a_{jk}$ (c_k étant une constante). Si la solution optimale $(a_{jk}^*)_{j=1..m}$ de (PP) vérifie $\bar{c}_k < 0$ alors $(a_{jk}^*)_{j=1..m}$ est une colonne améliorante et elle est ajoutée à A . Le (PMR) devient alors :

$$\begin{aligned} &\text{Minimiser } \left(\sum_{i=1}^n c_i x_i \right) + c_k x_k \\ &\text{sous } \forall j = 1, \dots, m, \left(\sum_{i=1}^n a_{ji} x_i \right) + a_{jk}^* x_k \geq b_j \\ &\quad \forall i = 1, \dots, n, x_i \geq 0, x_k \geq 0 \end{aligned}$$

Sinon (on a $\bar{c}_k \geq 0$) cela signifie qu'il n'existe pas de variable améliorante.

Cette méthode fonctionne lorsque le (PMR) est réalisable : il possède alors une solution duale qui permet la recherche d'une variable améliorante. Mais il existe des situations dans lesquelles le (PMR) peut ne pas être réalisable :

1. s'il est très difficile de trouver une solution (même de mauvaise qualité) à (\mathcal{P}) et donc d'initialiser \mathcal{V}' avec un ensemble de variables permettant d'avoir un (PMR) réalisable,
2. si la génération de colonnes est associée à un *Branch & Bound* dans le cadre de la résolution d'un problème impliquant des variables entières (on a alors un *Branch & Price*), le branchement peut rendre irréalisable le PL courant,

3. si la génération de colonnes est associée à des techniques de coupes (c'est le cas par exemple dans les algorithmes de *Branch Cut & Price*) et que ces nouvelles coupes rendent irréalisable le (PMR).

Dans ces situations il faut trouver un moyen d'ajouter des variables qui *tendent* à rendre le (PMR) réalisable. Pour cela on va utiliser le lemme de Farkas version matricielle.

2 Lemme de Farkas (version matricielle)

Je donne ici directement la version matricielle du lemme de Farkas qui va nous servir par la suite, pour une présentation plus détaillée voir [1] (version matricielle adaptée à la génération de colonnes) ou [2] (présentation générale des différentes versions du lemme de Farkas).

Soit A une matrice de $\mathcal{M}_{mn}(\mathbb{R})$, et $b \in \mathbb{R}^m$, alors :

Soit le système $Ax = b, x \geq 0$ admet une solution, **soit** il existe $y \in \mathbb{R}^m$ tel que $y^T A \leq 0$ et $y^T b > 0$

Les éléments du vecteur y sont appelés *multiplicateurs de Farkas*.

Pour rendre le système $Ax = b$ réalisable (s'il n'y est pas déjà !), il faut ajouter une colonne a à A telle que $y^T a > 0$. Pour cela il faut exhiber un vecteur y vérifiant $y^T A \leq 0$ et $y^T b > 0$ et résoudre l'inégalité $y^T a > 0$ (que l'on peut aussi écrire sous la forme $\sum_{j=1}^m y_j a_j > 0$ en notant $y = (y_j)_{j=1..m}$ et $a = (a_j)_{j=1..m}$) et dans laquelle le vecteur inconnu est a .

Dans la partie suivante on voit comment obtenir un vecteur y vérifiant le lemme de Farkas par un solveur linéaire.

3 Obtenir les valeurs des multiplicateurs de Farkas via un solveur linéaire

En pratique on considère un programme linéaire formé d'inégalités (et non d'égalités comme dans la définition théorique de la partie précédente mais on peut facilement s'y ramener en ajoutant des variables d'écart). On donne alors la définition suivante :

Pour un PL non faisable il existe un vecteur $y = (y_j)_{j=1..m}$ tel que l'inégalité $y^T Ax \geq y^T b$ est violée pour tout x et :

- $y_j \geq 0$ si la contrainte j est une contrainte " \geq "
- $y_j \leq 0$ si la contrainte j est une contrainte " \leq "

Ce vecteur y correspond au vecteur des multiplicateurs de Farkas. Les solveurs linéaires fournissent des valeurs pour ce vecteur. Par exemple :

1. Dans la *callable library* Cplex il existe la fonction :

```
int CPXdualfarkas(CPXCENVptr env, CPXCLPptr lp, double * y, double * p)
qui retourne, via le vecteur y, les valeurs de Farkas pour chacune des lignes du PL.
```

2. Dans SCIP il existe la fonction :

`SCIP_Real SCIPgetDualfarkasLinear (SCIP * scip, SCIP_CONS * cons)`
qui renvoie la valeur du multiplicateur de Farkas associé à la contrainte passée en paramètre.

Remarque : bien sûr si le PL est réalisable les multiplicateurs de Farkas n'existent pas !

4 Synthèse : génération d'une nouvelle colonne

On note $a = (a_j)_{j=1..m}$ la colonne que l'on souhaite générer. Dans le cas d'un PL irréalisable on a vu que l'on doit résoudre $\sum_{j=1}^m y_j a_j > 0$. On remarque qu'il s'agit d'une équation très proche de celle rencontrée dans la partie 1 (PP) pour trouver une variable améliorante : ici le multiplicateur de Farkas y_j joue le rôle de la variable duale λ_j et on ne prend pas en compte le coefficient dans la fonction objectif de la nouvelle variable éventuelle.

$\sum_{j=1}^m \lambda_j a_j > c$	$\sum_{j=1}^m y_j a_j > 0$
Cas PL réalisable : utilisation des variables duales $\lambda = (\lambda_j)_{j=1..m}$ et du coefficient objectif c de la nouvelle variable	Cas PL non réalisable : utilisation des multiplicateurs de Farkas $y = (y_j)_{j=1..m}$

Equation à résoudre pour générer une nouvelle colonne $a = (a_j)_{j=1..m}$ à la matrice des contraintes A .

5 Exemple : problème de découpe (*cutting stock problem*)

On reprend ici le fameux *cutting stock problem* (voir [4]) avec les données suivantes :

"Un atelier dispose de rouleaux de papier de 10 m. Les clients demandent des bandes de papier de longueur 2 m, 4 m, 8 m et 10 m en quantités respectives 2, 5, 20 et 15. Il faut donc découper les rouleaux de 10 m de sorte de satisfaire la demande tout en minimisant le nombre de rouleaux utilisés."

Il existe plusieurs manières de modéliser linéairement ce problème. On donne ici celle qui est adaptée à la génération de colonnes : pour cela on a besoin d'introduire la notion de *pattern* et de *coupe*. On appelle *pattern* la manière de découper un rouleau (cf exemples figure 1) et *coupe* une bande de papier de longueur donnée (ici 2, 4, 8 ou 10 m).

On note :

x_i : nombre de patterns i utilisés (un pattern requiert l'utilisation d'un rouleau)

n : nombre total de patterns

a_{ij} : nombre de coupes j dans le patterns i

d_j : demande en coupe j

Avec les données du problème on a donc 4 coupes différentes (2m, 4m, 8m et 10m) dont les demandes respectives sont $d_1 = 2, d_2 = 5, d_3 = 20, d_4 = 15$. Les patterns sont à définir, chaque manière de découper un rouleau fournit un pattern (voir quelques exemples figure 1). Etant donné la longueur du rouleau (10 m) les coupes dans un pattern i doivent vérifier l'équation :

$$2 a_{1i} + 4 a_{2i} + 8 a_{3i} + 10 a_{4i} \leq 10. \text{ (on ne coupe pas plus que la longueur disponible)}$$

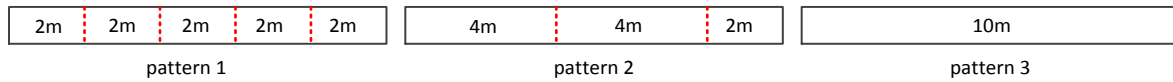


Figure 1 : exemples de patterns

En utilisant les notations précédentes on obtient le modèle linéaire :

$$\begin{aligned} & \text{Minimiser } \sum_{i=1}^n x_i \\ & \text{sous } \forall j \in \{1, \dots, 4\}, \sum_{i=1}^n a_{ij} x_i \geq d_j \\ & \quad x_i \in \mathbb{N} \end{aligned}$$

Notre but étant de générer des colonnes on va travailler sur le modèle relâché (avec les x_i dans \mathbb{R} et non dans \mathbb{N}), les contraintes d'intégrité se résolvant par exemple par branchement.

Si on utilise les patterns de la figure 1 on a :

$$n = 3, A = (a_{ij})_{1 \leq i \leq 3, 1 \leq j \leq 4} = \begin{pmatrix} 5 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

ce qui correspond au PL :

$$\begin{aligned} & \text{Minimiser } x_1 + x_2 + x_3 \\ & \text{sous } 5x_1 + x_2 \geq 2 \\ & \quad 2x_2 \geq 5 \\ & \quad 0 \geq 20 \\ & \quad x_3 \geq 10 \\ & \quad \forall i = 1, \dots, 3, x_i \in \mathbb{R} \end{aligned}$$

(P_{ex})

Ce PL est clairement infaisable puisqu'on n'a aucun pattern comprenant la coupe de 8m (d'où la contrainte $3 \cdot 0 \geq 20$). Pour ce PL Cplex nous fournit les valeurs suivantes pour les multiplicateurs de

$$\text{Farkas : } y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

On cherche donc une nouvelle colonne $(a_j)_{j=1..4}$ qui vérifie :

$$\sum_{j=1}^4 y_j a_j > 0 \quad (1)$$

$$2 a_1 + 4 a_2 + 8 a_3 + 10 a_4 \leq 10 \quad (2)$$

$$\forall j = 1, \dots, 4, \quad a_j \in \mathbb{N} \quad (3)$$

(1) : Equation donnée par Farkas pour générer une colonne qui tend à rendre le PL faisable

(2) : Contrainte liée à la définition d'une variable pour le *cutting stock problem* (la variable doit définir un pattern)

(3) : a_j correspond au nombre de coupes de type j qui sera dans le pattern, c'est donc forcément un entier.

Il est clair que $a_3 = 1, a_1 = a_2 = a_4 = 0$ est solution. On peut donc ajouter la colonne $\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ à la

matrice des contraintes (elle est ici identique au vecteur des multiplicateurs de Farkas mais c'est une coïncidence, le problème choisi étant extrêmement simple). Avec cette nouvelle variable le problème (P_{ex}) devient :

(P_{ex}) après génération de la nouvelle colonne	Minimiser	$x_1 + x_2 + x_3 + x_4$
	sous	$5 x_1 + x_2 \geq 2$
		$2 x_2 \geq 5$
		$x_4 \geq 20$
		$x_3 \geq 10$
		$\forall i = 1, \dots, 4, \quad x_i \in \mathbb{R}$

Références

[1] G. Gamrath, Generic Branch-Cut-and-Price, 2010 :

http://www.zib.de/gamrath/publications/gamrath2010_genericBCP.pdf

[2] http://fr.wikipedia.org/wiki/Lemme_de_Farkas

[3] Jacques Teghem. Programmation linéaire. Editions de l'université de Bruxelles, Editions Ellipses. 2004

[4] http://en.wikipedia.org/wiki/Cutting_stock_problem