RAIRO. RECHERCHE OPÉRATIONNELLE

PHAM DINH TAO

Un algorithme pour la résolution du programme linéaire général

RAIRO. Recherche opérationnelle, tome 25, n° 2 (1991), p. 183-201

<http://www.numdam.org/item?id=RO_1991__25_2_183_0>

Recherche opérationnelle/Operations Research
(vol. 25, n° 2, 1991, p. 183 à 201)

UN ALGORITHME POUR LA RÉSOLUTION DU PROGRAMME LINÉAIRE GÉNÉRAL (*)

par PHAM DINH TAO (1)

Résumé. — *Un algorithme appelé « simplexe généralisé, gradient projeté » ((SGGP) est proposé pour résoudre le programme linéaire général. Cet algorithme travaille directement dans l'espace des variables et comporte deux phases. La phase I (initialisation de l'algorithme ou recherche d'un sommet d'un polyèdre convexe qui se sert de la phase II comme dans l'algorithme du simplexe) est construite sous deux formes (procédé primal et procédé dual). Conçu comme une extension naturelle de l'algorithme du simplexe, l'algorithme SGGP englobe ses variantes usuelles (algorithme dual du simplexe, algorithme pour le programme linéaire avec variables bornées, ...).*

Enfin l'algorithme SGGP est l'algorithme du gradient projeté avec comme point de départ un sommet du polyèdre convexe.

Mots clés : Programme linéaire général; algorithme simplexe; algorithme du gradient projeté.

Abstract. — *We propose an algorithm, called SGGP, for solving a general linear programming problem. The proposed algorithm works directly in the primal variables space and comprises two phases. Phase I (initialization of the algorithm or determination of an extreme point of a convex polyhedron, which uses phase II as in the Simplex Algorithm) has been constructed in two forms (primal and dual procedures). Designed as a natural extension of the Simplex Algorithm, the algorithm SGGP contains its usual variants (dual Simplex, case of bounded variables, ...).*

We show that SGGP is a projected gradient method with an extreme point of the convex polyhedron as the starting point.

Keywords : General linear programming; Simplex Algorithm; Projected gradient method.

1. INTRODUCTION

Nous présentons dans ce papier un algorithme appelé SGGP (Simplexe Généralisé, Gradient Projeté) pour la résolution du programme linéaire général : Max $\{cx : Ax = b,\ Bx \leqq a\}$. Cet algorithme, plutôt situé dans le cadre

(*) Reçu juin 1989.
(1) Équipe Modélisation et Optimisation, Laboratoire T.I.M.3, I.M.A.G., B.P. n° 53X, 38401 Grenoble Cedex.

Recherche opérationnelle/Operations Research, 0399-0559/91/02 183 19/$ 3.90

de l'analyse et de l'optimisation convexes, est fondamentalement basé sur les conditions d'optimalité et sur la caractérisation d'un sommet de $K=\{x\in\mathbb{R}^n: Ax=b,\ Bx\leqq a\}$ comme étant un point de K qui admet n contraintes actives linéairement indépendantes. Cet algorithme n'introduit pas les variables d'écart, on travaille directement dans l'espace des variables x.

La cohérence et l'unité de l'algorithme SGGP résident (comme dans l'algorithme du simplexe) dans le fait suivant : la phase I utilise la phase II. Nous présentons ici deux nouvelles techniques de construction de la phase I (procédé primal et procédé dual).

Le procédé primal consiste en l'application de la phase II à une suite finie de programmes linéaires (dans le même espace primal $\mathbb{R}^n$) dont on connaît un sommet, solution réalisable. Ces programmes linéaires seront définis successivement en tenant compte des contraintes restant violées de K au sommet solution du programme linéaire précédent. On arrive ainsi à obtenir un sommet de K ou la conclusion de la vacuité de K.

Le procédé dual, par contre ne résout qu'un seul programme linéaire. Son principe est très simple : on calcule un point défini par n contraintes linéairement indépendantes de K, (en général par la méthode d'élimination de Gauss par exemple) puis on définit le vecteur coût d (à partir des contraintes explicitées) de manière que le problème primal auxiliaire suivant soit dual réalisable (*i. e.* un sommet solution réalisable de son dual est explicité) :

$$\left.\begin{aligned} Ax&=b\\ Bx&\leqq a\\ dx&=z\,(\max).\end{aligned}\right\} \qquad \text{(PA)}$$

L'application de la phase II de l'algorithme SGGP à ce problème dual auxiliaire conduit à un sommet de K (de surcroît solution du problème primal auxiliaire) ou à la conclusion de la vacuité de K.

Ces deux procédés éclairent sous un nouveau jour la recherche des techniques appropriées de détermination d'un sommet d'un polyèdre convexe. L'application de SGGP au programme linéaire mis sous forme standard par rapport à une base réalisable nous redonne l'algorithme révisé du simplexe. Plus exactement, la description de l'algorithme du simplexe est la forme simplifiée de SGGP dans laquelle la résolution des systèmes linéaires (intervenant dans le calcul de la direction de déplacement, le pas de déplacement et la vérification des conditions d'optimalité) se fait via la technique de décomposition de ces systèmes (utilisant la structure creuse de la matrice représentant les contraintes de non négativité de la variable x). De même

l'algorithme dual du simplexe est exactement l'algorithme SGGP appliqué au problème dual dont le primal est dual réalisable.

D'autre part la présentation de cet algorithme est particulièrement claire et constructive dans les programmes linéaires avec bornes. Enfin ce qui est fondamental est que l'algorithme SGGP n'est rien d'autre que l'algorithme du gradient projeté appliqué au programme linéaire général lorsque le point de départ est un sommet du polyèdre K. Ce point de vue original permet de situer l'algorithme du simplexe dans le cadre général de l'optimisation convexe.

Certains problèmes d'application qui étaient à l'origine de ce travail sont présentés ainsi que la description simplifiée de la phase I dans les programmes linéaires sous forme standard et canonique.

Des expérimentations numériques comparatives sont exposées dans [9, 16]. Ces résultats montrent que SGGP est plus rapide que l'algorithme révisé du simplexe : le rapport du temps total de calcul (phase I et phase II) varie entre 1 et 3.5. La rapidité est encore accrue pour la phase I dont le rapport varie entre 3 et 8.

Ces résultats sont encourageants, ils prouvent l'utilité et la performance de l'algorithme SGGP.

Une étude comparative complète de ces algorithmes est en cours, dans laquelle le procédé de la « forme produit de l'inverse » sera utilisé pour optimiser les calculs dans les résolutions des systèmes linéaires.

2. NOTATIONS. DÉFINITIONS ET PROPRIÉTÉS FONDAMENTALES DE LA PROGRAMMATION LINÉAIRE

Ce paragraphe contient des notations élémentaires et des résultats fondamentaux de la programmation linéaire qui sont nécessaires à notre travail. Soit A une matrice de type $m \times n$. Soient I et J deux parties de $\{1, \ldots, m\}$ et de $\{1, \ldots, n\}$ respectivement. On note A_I la matrice formée par les lignes $A_{ra(i)}$, $i = 1, \ldots, |I|$ ($|I|$ désigne le cardinal de I) rangées dans cet ordre, ra étant l'application injective de $\{1, \ldots, |I|\}$ dans $\{1, \ldots, m\}$ qui correspond d'une façon biunivoque à $I : I = \{ra(1), \ldots, ra(|I|)\}$. De même A^J désigne la matrice formée par les colonnes $A^{ra(j)}$, $j = 1, \ldots, |J|$, rangées dans cet ordre. Le complémentaire de I dans $\{1, \ldots, m\}$ est noté I^*.

Considérons maintenant le polyèdre K défini par :

$$K=\{x\in\mathbb{R}^n : Ax=b,\ Bx\leqq a\}$$

où A est une matrice de type $m\times n$, de rang m et B est une matrice de type $p\times n$. On dit que x est un sommet de K si x est la solution d'un système linéaire :

$$Ax=b \tag{1}$$

$$B_I x=a_I \tag{2}$$

telle que

$$B_{I'}x\leqq a_{I'} \tag{3}$$

où I est un sous-ensemble de $\{1, \ldots, p\}$, $|I|=n-m$, tel que les lignes de A et celles de B_I sont linéairement indépendantes. L'ensemble des sommets de K sera noté $V(K)$. Il s'ensuit que la non vacuité de $V(K)$ implique que la matrice $\begin{bmatrix} A \\ B \end{bmatrix}$ (formée par les lignes de A et B) est de rang n. Une contrainte $B_i x\leqq a_i$ est dite active en $x\in K$ si $B_i x=a_i$. Pour tout $x\in K$, on note $\mathscr{K}(x)=\{i=1, \ldots, p : B_i x=a_i\}$.

Un sommet x de K est dit non dégénéré si $|\mathscr{K}(x)|=n-m$, dans le cas contraire x est dit dégénéré. Un sommet x de K est non dégénéré si et seulement si il existe une partie I unique telle que x est la solution de (1), (2), (3). Deux sommets x et x' de K sont dits adjacents s'ils sont les solutions de (1) et (2) avec I et I' tels que $|I\cap I'|=n-(m+1)$.

Conditions d'optimalité d'un programme linéaire général sous forme canonique

Considérons maintenant le programme linéaire général (P) sous sa forme canonique

$$\left.\begin{aligned} Ax&=b \\ Bx&\leqq a \\ cx&=z\,(\max)\end{aligned}\right\} \tag{P}$$

Nous rappelons le résultat bien connu suivant [4, 11, 14, 15].

THÉORÈME 2 : 1. $x \in K$ est une solution de (P) si et seulement si il existe un m-vecteur ligne λ et un p-vecteur ligne $\mu \geqq 0$ tels que :

$$c = \lambda A + \mu B \tag{4}$$

$$\mu (Bx - a) = 0 \tag{5}$$

2. En particulier soit $x \in V(K)$ défini par (1), (2) et (3) et soit (λ, μ^I) la solution du système linéaire :

$$(\lambda, \mu^I) \begin{bmatrix} A \\ B_I \end{bmatrix} = c \tag{6}$$

Alors x est une solution de (P) si $\mu^I \geqq 0$. Cette condition suffisante est aussi nécessaire si $x \in V(K)$ est non dégénéré. ■

3. ALGORITHME POUR LA RÉSOLUTION DU PROBLÈME (P)

Nous commençons par indiquer comment calculer un sommet u de K adjacent à un sommet v donné.

3.1. Détermination d'un sommet adjacent

Le processus est très simple : Soit $v \in V(K)$ déterminé par (1), (2) et (3). Soit $i \in \{1, \ldots, |I|\}$ fixé, on pose $I(i) = I \setminus \{ra(i)\}$. Considérons le système linéaire suivant :

$$Ax = 0 \tag{7}$$

$$B_{I(i)} x = 0 \tag{8}$$

$$B_{ra(i)} x = -1 \tag{9}$$

qui admet une solution unique $d^{(i)}$.

Soit $v^{(i)} = v + \lambda_i d^{(i)}$. On a $A v^{(i)} = b$, $B_{I(i)} v^{(i)} = a_{I(i)}$ et $B_{ra(i)} v^{(i)} = a_{ra(i)} - \lambda_i$. Par suite $B_{ra(i)} v^{(i)} \leqq a_{ra(i)}$ si et seulement si $\lambda_i \geqq 0$.

Voyons les autres conditions portant sur λ_i pour assurer que $v^{(i)} \in K$. Il est clair que $v^{(i)} \in K$ si et seulement si $\lambda_i \geqq 0$ vérifie :

$$B_{I^*} v^{(i)} = B_{I^*} v + \lambda_i B_{I^*} d^{(i)} \leqq a_{I^*}.$$

Puisque $B_{I^*} v \leqq a_{I^*}$ on a les propriétés suivantes :

(i) Si $B_{I^*} d^{(\iota)} \leqq 0$ alors $v^{(\iota)} = v + \lambda_\iota d^{(\iota)} \in K$ pour tout $\lambda_\iota \geqq 0$. Dans ce cas il n'y a pas de sommet adjacent à v dans la direction $d^{(\iota)} : \{ v + \lambda_\iota d^{(\iota)} : \lambda_\iota \geqq 0 \}$ est un rayon extrémal de K issu de v.

(ii) Sinon l'ensemble $C(I, \iota) = \{ k \in I^* : B_k d^{(\iota)} > 0 \}$ est non vide et alors $v^{(\iota)} = v + \lambda_\iota d^{(\iota)} \in K$ si et seulement si :

$$\lambda_\iota \leqq \min \left\{ \frac{a_k - B_k v}{B_k d^{(\iota)}} : k \in C(I, i) \right\} = \lambda_\iota^* \tag{10}$$

Dans ce cas $v^{(\iota)} = v + \lambda_\iota^* d^{(\iota)}$ est un sommet de K adjacent à v. En effet pour tout $k \in C(I, i)$, B_k, les lignes de A et celles de $B_{I(\iota)}$ sont linéairement indépendante. Par suite si $k \in C'(I, i) = \{ k \in C(I, i) : (a_k - B_k v)/B_k d^{(\iota)} = \lambda_\iota^* \}$ alors :

$$B_J v^{(\iota)} = a_J$$

et

$$B_{J^*} v^{(\iota)} \leqq a_{J^*}$$

où

$$J = I(i) \cup \{ k \}, \qquad ra(\iota) = k.$$

Autrement dit $v^{(\iota)} \in V(K)$ qui est défini par (1), (2) et (3) dans lesquels I est remplacé par J. De plus si $\lambda_\iota^* > 0$ alors $v^{(\iota)}$ est différent de v et on a : $\mathscr{K}(v^{(\iota)}) = I(i) \cup C'(I, i)$. Il s'ensuit qu'au cas où $\lambda_\iota^* > 0$, $v^{(\iota)}$ est un sommet non dégénéré si et seulement si $|C'(I, i)| = 1$.

Remarquons enfin que si v est non dégénéré alors $\lambda_\iota^* > 0$ et que dans ce cas il y a exactement $(n - m)$ sommets adjacents à v que l'on peut tous déterminer.

3.2. Description de la phase II de l'algorithme SGGP pour la résolution de (P)

La phase II de SGGP suppose connu un sommet de K qui sera fourni par la phase I présentée dans le paragraphe 5. Sa description est très simple :

Il suffit d'indiquer, lorsqu'un sommet courant v de K n'est pas une solution de (P) (ou plutôt ne vérifie pas la condition suffisante 2 du théorème 2), comment passer à un sommet adjacent afin d'augmenter la valeur de la fonction objective cx. Soit $v \in V(K)$ défini par (1), (2) et (3). On résout (6) pour obtenir μ^I. Si $\mu^I \geqq 0$ alors v est une solution de (P). Sinon soit $i \in \{ 1, \ldots, |I| \}$ tel que $\mu^{ra(\iota)} < 0$ et soit $d^{(\iota)}$ la solution du système linéaire

(7), (8) et (9). Tenant compte de (6), (7), (8) et (9) on a :

$$cd^{(i)} = (\lambda A + \mu^I B_I)\, d^{(i)} = \mu^{ra\,(i)} B_{ra\,(i)}\, d^{(i)} = -\mu^{ra\,(i)}.$$

Par suite le déplacement de v à $v+\lambda_i d^{(i)}$ avec $\lambda_i \geqq 0$ augmente la valeur de cx de $-\lambda_i \mu^{ra\,(i)}$. Comme dans le paragraphe 3.1 on va distinguer deux cas :

(i) $B_{I^*} d^{(i)} \leqq 0$ alors $v+\lambda_i d^{(i)} \in V(K)$ pour tout $\lambda_i \geqq 0$. Dans ce cas on a :

$$\lim_{\lambda_i \to +\infty} \{ cx : x = v + \lambda_i d^{(i)} \} = +\infty$$

et le problème (P) n'admet pas de solution finie.

(ii) Sinon $C(I, i) = \{ k \in I^* : B_k d^{(i)} > 0 \} \neq \varnothing$ et $v^{(i)} = v + \lambda_i^* d^{(i)} \in K$ avec λ_i^* défini par (10), est un sommet de K adjacent à v qui sera le prochain sommet courant dans l'algorithme SGGP.

La convergence de l'algorithme SGGP résulte de la stricte augmentation de la fonction objective (au moins en l'absence de la dégénérescence) et la finitude de $V(K)$.

THÉORÈME 3 (convergence) : *Si aucun sommet généré par cet algorithme n'est dégénéré alors après un nombre fini d'itérations il permet d'obtenir une solution de* (P) *ou de déclarer que* (P) *n'a pas de solution.* ■

Il est bien connu [15] que la règle de Bland ne dépend pas du type de contraintes (inégalités) linéaires ($Bx \leqq a$ ou $x \geqq 0$), elle peut être ainsi incorporée à l'algorithme SGGP présenté ci-dessus :

1. On choisit $i \in \{1, \ldots, |I|\}$ tel que $ra(i)$ est le plus petit élément de I vérifiant $\mu^{ra\,(i)} < 0$ (dans le cas où $\mu^I \geqq 0$ n'a pas lieu).

2. Dans (ii) on choisit k comme étant le plus petit élément de I^* tel que $B_k d^{(i)} > 0$.

4. RELATION AVEC L'ALGORITHME DU SIMPLEXE ET L'ALGORITHME DUAL DU SIMPLEXE

4.1. Lien avec l'algorithme du simplexe

Considérons maintenant le programme linéaire (P') sous forme standard

$$\left.\begin{array}{l} Ax = b, \qquad x \geqq 0 \\ cx = z\,(\max) \end{array}\right\} \qquad (P')$$

où A est une matrice de type $m \times n$ de rang m. (P') est un cas particulier de (P) où $p=n$, $a=0$ et $B=-U_n$, (U_n étant la matrice identité d'ordre n).

On va montrer que l'application de l'algorithme SGGP au problème (P') permet de retrouver l'algorithme du simplexe. Soit

$$K'=\{x\in\mathbb{R}^n : Ax=b,\ x\geqq 0\}.$$

Voyons tout d'abord comment introduire les bases réalisables de (P') et caractériser les sommets de K' à l'aide de ces bases.

Comme dans le paragraphe 2, $x\in V(K')$ si et seulement si x est la solution de (1), (2) et (3) avec ici $B=-U_n$.

On a donc :

$$x_I=0 \tag{11}$$

Et le système linéaire (1) devient :

$$Ax=A^I x_I+A^{I^*}x^{I^*}=b$$
$$A^{I^*}x_{I^*}=b \tag{12}$$

On en déduit facilement le résultat classique sur la caractérisation des sommets de K' [4, 13, 15]

THÉORÈME 4 : *Soit* $K'=\{x\in\mathbb{R}^n : Ax=b,\ x\geqq 0\}$ *avec* A *une matrice de type* $m\times n$, *de rang* m. *Alors* $x\in V(K')$ *si et seulement si il existe* $J\subset\{1,\ldots,n\}$; $|J|=m$, *tel que :*

(i) A^J *est non singulière et* $(A^J)^{-1}b\geqq 0$

(ii) $x_J=(A^J)^{-1}b$ *et* $x_{J^*}=0$. ■

Une telle partie J est usuellement appelée, en programmation linéaire, base réalisable de (P') et le sommet x défini par (i) et (ii) solution réaliable de base J. On remarque que J et J^* ne sont autres que I^* et I respectivement dans (11) et (12). Dès lors l'équation (6) devient :

$$c^I=\lambda A^I-\mu^I$$
$$c^{I^*}=\lambda A^{I^*}$$

car

$$(\mu^I B_I)^I=-\mu^{I^*} \qquad \text{et} \qquad (\mu^I B_I)^{I^*}=0.$$

On en déduit que $-\mu^{J^*}=\mu^I=c^{J^*}-c^J(A^J)^{-1}A^{J^*}$.

Ceci étant, l'algorithme SGGP appliqué à (P') redonne exactement l'algorithme du simplexe [4, 13, 15].

Remarques importantes

Les systèmes linéaires intervenant dans le calcul de la direction de déplacement, le calcul du pas de déplacement et la vérification des conditions d'optimalité deviennent plus simple dans la programmation linéaire sous forme standard. C'est sous cette forme simplifiée de SGGP que l'on retrouve l'algorithme du simplexe.

Remarquer que cette simplification est due à la technique de décomposition des systèmes linéaires précédents. Pour plus de détails, *voir* [9], [16].

Cette technique de décomposition sera d'une utilisation constante dans l'implémentation de différentes versions de l'algorithme SGGP, notamment dans la phase *I* de SGGP appliqué aux programmes linéaires sous forme canonique et standard [9, 16].

4.2. Lien avec l'algorithme dual du simplexe

Considérons un programme linéaire sous forme standard :

$$\left.\begin{array}{c} Ax=b, \qquad x\geqq 0 \\ cx=z\,(\max) \end{array}\right\} \tag{P'}$$

On suppose que le rang de la matrice A, de type $m\times n$, est égal à m et que le problème (P') est dual réalisable, c'est-à-dire que l'on dispose d'un ensemble $J\subset\{1, \ldots, n\}$, $|J|=m$ tel que :

(i) A^J est non singulière.

(ii) $c^{J^*}-\pi A^{J^*}\leqq 0$ où $\pi=c^J(A^J)^{-1}$

L'ensemble J est une base du système linéaire $Ax=b$ qui n'est pas nécessairement réalisable pour (P'). Le problème dual (D) s'écrit :

$$\left.\begin{array}{c} yA\geqq c \\ yb=w\,(\min) \end{array}\right\} \tag{D}$$

Il est clair que π est une solution réalisable de (D), et plus précisément un sommet du polyèdre convexe des solutions réalisables de (D).

En appliquant à (D) l'algorithme SGGP décrit dans le paragraphe 3, on retrouve l'algorithme dual du simplexe [4, 13, 15].

4.3. Programme linéaire avec contraintes de bornes

Ce sont des problèmes de ce type

$$\left.\begin{array}{c} Ax=b \\ a_I \leqq x_I \leqq b_I \\ cx=z\,(\max) \end{array}\right\} \qquad (P)$$

où I est une partie $\{1, \ldots, n\}$. Les constantes a_i peuvent être égales à $-\infty$ et les constantes b_i peuvent être égales à $+\infty$. Des méthodes issues de l'algorithme du simplexe sont adaptées à la résolution de ce type de problème. Mais très souvent leur présentation (plus compliquées que l'algorithme du simplexe et l'algorithme dual du simplexe) ne sont pas assez naturelles pour faciliter leur compréhension en profondeur.

Par contre en traitant toutes les contraintes linéaires (y compris les contraintes de bornes) de (P) de la même manière, c'est-à-dire qu'en appliquant l'algorithme SGGP au problème (P), on obtient une description claire, simple et facile à comprendre (*cf.* § 3). D'autres techniques pour traiter le programme linéaire avec contraintes de bornes sont précisées dans le paragraphe 6.4.

Le problème (P) est un cas particulier du programme linéaire général auquel l'application de l'algorithme SGGP permet de retrouver les algorithmes connus [4, 13, 15] et surtout de remédier aux inconvénients cités ci-dessus.

4.4. Lien avec la méthode du gradient projeté

Donnons tout d'abord une description succinte de la méthode du gradient projeté dû à Rosen [4] pour la résolution du problème d'optimisation suivant :

$$\left.\begin{array}{c} \mathrm{Min} f(x) \\ Ax=b \\ Bx \leqq a \end{array}\right\} \qquad (Q)$$

où A et B sont les matrices vérifiant les mêmes hypothèses que dans le paragraphe 2 et $f(x)$ est une fonction différentiable.

Cette méthode consiste en la construction d'une suite (x^k) de solutions admissibles de la manière suivante : On suppose la régularité des contraintes en tout point x^k, *i.e.* les lignes de A et les lignes de $B_{\mathscr{K}(x^k)}$ sont linéairement

indépendantes. Le passage de x^k à x^{k+1} est défini par :

(i) La direction de déplacement x^k qui est la projection orthogonale de $-\nabla f(x^k)$ sur le sous-espace tangent $V_k = \{ : \mathrm{Ad} = 0, B_{\mathscr{K}(x^k)} d = 0 \}$ de dimension $n - (m + |\mathscr{K}(x^k)|)$.

(ii) Si $d^k \neq 0$, on calcule α^k et β^k solutions respectivement de

$$\mathrm{Max}\{\alpha : x^k + \alpha d^k \in K\}, \qquad K = \{x \in \mathbb{R}^n : A x = b, B x \leqq a\}$$

$$\mathrm{Min}\{f(x^k + \beta d^k) : 0 \leqq \beta \leqq \alpha^k\}$$

On pose $x^{k+1} = x^k + \beta^k d^k$ et réitère le processus.

(iii) Si $d^k = 0$ alors le système $\begin{bmatrix} A \\ B_{\mathscr{K}(x^k)} \end{bmatrix}^{\mathrm{T}} \gamma = M_k^{\mathrm{T}} \gamma = -\nabla f(x^k)$ admet une solution unique, plus précisément : $\gamma = -(M_k M_k^{\mathrm{T}})^{-1} M_k \nabla f(x^k)$.

(*a*) Si $\gamma_j \geqq 0$ pour tout j correspondant aux contraintes définies par $B_{\mathscr{K}(x^k)}$ alors x^k vérifie les conditions de Kuhn-Tucker relatives au problème (Q) et par suite est une solution de (Q) si f est convexe.

(*b*) Sinon on choisit un $\gamma_k < 0$ qui correspond à une contrainte active en x^k définie par une ligne de $B_{\mathscr{K}(x^k)}$. Soit $\mathscr{K}'(x^k) = \mathscr{K}(x^k) \setminus \{\text{indice de cette ligne}\}$. Retourner à (i) avec $\mathscr{K}'(x^k)$ à la place de $\mathscr{K}(x^k)$. Soit maintenant $f(x) = cx$ et appliquons la méthode du gradient projeté avec un vecteur initial $x^0 \in V(K)$, il est clair que $V_0 = \{0\}$ et par suite $d^0 = 0$. L'étape (iii) correspond exactement à la résolution du système linéaire (6) car M_k est une matrice carrée non singulière. Dans les itérations suivantes le calcul de d^k, $k \geqq 1$, est immédiat car $\dim V_k = 1$, V_k est exactement une droite parallèle à une arête de K issue x^k. De plus il est clair que $\alpha^k = \beta^k$. Finalement si $f(x) = cx$ et si l'on part d'un $x^0 \in V(K)$, la méthode du gradient projeté donne exactement l'algorithme SGGP.

5. INITIALISATION DE L'ALGORITHME SGGP (PHASE I). DÉTERMINATION D'UN SOMMET DE K PAR L'ALGORITHME SGGP

Ce chapitre est consacré à l'initialisation de l'algorithme SGGP pour la résolution du programme linéaire général (*cf.* § 3).

$$\left.\begin{aligned} A x &= b \\ B x &\leqq a \\ cx &= z\,(\max) \end{aligned}\right\} \qquad (P)$$

i.e. à la détermination d'un sommet de $K=\{x\in\mathbb{R}^n : Ax=b,\ Bx\leqq a\}$, A étant une matrice de type $m\times n$ et B une matrice de type $p\times n$. Et comme dans l'algorithme du simplexe, la phase I de SGGP n'est autre que sa phase II appliquée à des programmes linéaires convenablement définis.

Nous allons présenter ci-dessous deux nouveaux procédés d'initialisation. On peut supposer, sans perte de généralité, que le rang de A est égal à m. Si tel n'est pas le cas, il suffirait d'appliquer la méthode d'élimination de Gauss au système $Ax=b$ pour enlever les lignes redondantes lorsque les équations du système $Ax=b$ sont compatibles ou arriver à la conclusion de la vacuité de K dans le cas contraire.

5.1. Procédé primal

On recherche un ensemble $I_0=\{ra(1), \ldots, ra(|I_0|)\}$ de $\{1, \ldots, p\}$ tel que $|I_0|=n-m$ et que le rang de $\begin{bmatrix} A \\ B_{I_0} \end{bmatrix}$ soit égal à n (*cf.* § 3.1).

On pourrait, pour cela, utiliser la méthode d'élimination de Gauss opérant sur les lignes de A et de B. Si un tel ensemble I_0 n'existe pas, $V(K)=\varnothing$.

Le principe de ce procédé est très simple : On résout le système linéaire

$$\begin{bmatrix} A \\ B_{I_0} \end{bmatrix} x=\begin{bmatrix} b \\ a_{I_0} \end{bmatrix}$$

pour obtenir la solution $x^{(0)}$ qui est un sommet du polyèdre

$$K(I_0)=\{x : Ax=b,\ B_{I_0}x\leqq a_{I_0},\ B_{\mathscr{J}(x^{(0)})}x\leqq a_{\mathscr{J}(x^{(0)})}\}$$

où :

$$\mathscr{J}(x^{(0)})=\{i\in I_0^* : B_i x^{(0)}\leqq a_i\}$$

Si $\mathscr{J}^*(x^{(0)})=I_0^*\setminus\mathscr{J}(x^{(0)})=\varnothing$ alors $x^{(0)}\in V(K)$.

Dans le cas contraire on va construire une suite finie de programmes linéaires auxquels la phase II de l'algorithme SGGP peut être applicable.

On obtiendra ainsi une suite finie (x^k) (de sommets solutions desdits programmes linéaires), le passage de x^k à x^{k+1} augmente d'au moins une unité le nombre des contraintes de K qui sont vérifiées. Ainsi après la résolution d'un nombre fini de programmes linéaires, ce procédé permet d'obtenir un sommet de K ou de conclure que K est vide.

Voici donc la description du procédé primal :

0. On résout le système linéaire

$$\begin{bmatrix} A \\ B_{I_0} \end{bmatrix} x = \begin{bmatrix} b \\ a_{I_0} \end{bmatrix}$$

pour obtenir la solution $x^{(0)}$.

1. Si $\mathcal{J}^*(x^{(0)}) = \{ i \in I_0^* : B_i x^{(0)} > a_i \} = \varnothing$ alors $x^{(0)} \in V(K)$.

2. Sinon on choisit $i_0 \in \mathcal{J}^*(x^{(0)})$, par exemple i_0 défini par :

$$B_{i_0} x^{(0)} - a_{i_0} = \operatorname{Max} \{ B_i x^{(0)} - a_i : i \in \mathcal{J}^*(x^{(0)}) \}.$$

Soit :

$$\left.\begin{array}{c} Ax = b \\ B_{I_0} x \leqq a_{I_0} \\ B_{\mathcal{J}(x^{(0)})} x \leqq a_{\mathcal{J}(x^{(0)})} \\ -B_{i_0} x = z\,(\max). \end{array}\right\} \qquad (P(I_0))$$

On a $x^{(0)} \in V(K(I_0))$ par suite on peut appliquer la phase II de SGGP à la résolution de $(P(I_0))$. On résout le système linéaire

$$(\lambda, \mu^{I_0}) \begin{bmatrix} \mathrm{A} \\ \mathrm{B}_{I_0} \end{bmatrix} = -B_{i_0}$$

2.1. Si $\mu^{I_0} \geqq 0$ alors $x^{(0)}$ est une solution de $(P(I_0))$ (*cf.* § 2). Dans ce cas on a $-B_{i_0} x < -a_{i_0}$, $\forall x \in K(I_0)$.

Autrement dit $K = \varnothing$ car $K \subset K(I_0)$.

2.2. Sinon on choisit $j_0 \in \{ 1, \ldots, |I_0| \}$ tel que $\mu^{ra(j_0)} < 0$, par exemple :

$$\mu^{ra(j_0)} = \min \{ \mu^{ra(i)} : i \in \{ 1, \ldots, |I_0| \} \}$$

On calcule $d^{(j_0)}$ solution du système linéaire (*cf.* § 3.2)

$$\mathrm{Ad} = 0$$

$$B_{I_0(j_0)} d = 0$$

$$B_{ra(j_0)} d = -1$$

2.2.1. Si $\mathcal{J}(x^{(0)})$ est vide ou si $B_{\mathcal{J}(x^{(0)})} d^{(j_0)} \leqq 0$ alors $x^{(0)} + \lambda d^{(j_0)} \in K(I_0)$ pour tout $\lambda \geqq 0$: l'ensemble $\{ x^{(0)} + \lambda d^{(j_0)} : \lambda \geqq 0 \}$ est un rayon extrémal de $K(I_0)$ le

long duquel la fonction objective $-B_{i_0}x$ augmente indéfiniment. Dans ce cas on calcule λ_{j_0} tel que : $-B_{i_0}(x^{(0)}+\lambda_{j_0}d^{(j_0)})=-a_{i_0}$.

Soit

$$\lambda_{j_0}=\frac{a_{i_0}-B_{i_0}(x^{(0)})}{B_{i_0}d^{(j_0)}}>0 \tag{13}$$

(Rappelons ici que $B_{i_0}d^{(j_0)}=\mu^{ra(j_0)}<0$ (*cf.* § 3.2) et $B_{i_0}x^{(0)}>a_{i_0}$).

On pose $I_1=[I_0\setminus\{ra(j_0)\}]\cup\{i_0\}$, $ra(j_0)=i_0$, $x^{(1)}=x^{(0)}+\lambda_{j_0}d^{(j_0)}$.

La contrainte $-B_{i_0}x\leqq -a_{i_0}$ qui est violée par $x^{(0)}$ devient active en $x^{(1)}$.

On remplace I_0 par I_1 et $x^{(0)}$ par $x^{(1)}$ et on retourne à l'étape 1.

2.2.2. Sinon, on calcule

$$\lambda^*_{j_0}=\text{Min}\left\{\frac{a_i-B_ix^{(0)}}{B_id^{(j_0)}}: i\in\mathscr{I}(x^{(0)}),\ B_id^{(j_0)}>0\right\}$$

. On choisit $i_1\in\mathscr{I}(x^{(0)})$ tel que $B_{i_1}d^{(j_0)}>0$ et $(a_{i_1}-B_{i_1}(x^{(0)}))/B_{i_1}d^{(j_0)}=\lambda^*_{j_0}$.

On pose $I_1=[I_0\setminus\{ra(j_0)\}]\cup\{i_1\}$, $ra(j_0)=i_1$, $x^{(1)}=x^{(0)}+\lambda^*_{j_0}d^{(j_0)}$.

On remplace I_0 par I_1 et $x^{(0)}$ par $x^{(1)}$ et on retourne à l'étape 1.

Dans le cas des programmes linéaires sous forme standard ou canonique, on obtiendra des versions simplifiées du procédé primal [9, 16].

5.2. Procédé dual

Ce deuxième procédé est basé sur l'application de notre algorithme à un problème dual dont le primal est construit de façon que les conditions suivantes soient vérifiées :

(i) Le problème primal est dual réalisable

(ii) Tout sommet solution du problème primal est un sommet de K.

La définition de la duale-réalisabilité d'un programme linéaire général sera donnée dans le paragraphe 6.3.

Comme dans le procédé primal, on détermine une matrice B_I telle que le rang de $\begin{bmatrix} A \\ B_I \end{bmatrix}$ soit égal à n.

Soit $d=(\lambda, \mu^I)\begin{bmatrix} A \\ B_I \end{bmatrix}=\lambda A+\mu^I B_I$, où λ est un m-vecteur ligne quelconque et μ un p-vecteur ligne tel que $\mu^I>0$ et $\mu^{I^*}=0$.

Considérons les programmes linéaires auxiliaires duaux suivants :

$$\left.\begin{aligned} Ax &= b \\ Bx &\leqq a \\ dx &= z\,(\max) \end{aligned}\right\} \qquad (P)$$

$$\left.\begin{aligned} yA + vB &= d, v \geqq 0 \\ yb + va &= w\,(\min). \end{aligned}\right\} \qquad (DA)$$

Il est clair que (λ, μ) est un sommet du polyèdre convexe des solutions réalisables de (DA). Autrement dit (PA) est dual réalisable et l'application de notre algorithme (phase II) au problème dual (DA) va fournir un sommet de K, solution de (PA) ou bien conduire à la conclusion de la vacuité de K [en cas de non existence de solution de (DA)].

Comme dans le procédé primal, on obtient des versions simplifiées du procédé dual pour les programmes linéaires sous forme standard ou canonique [9, 16].

6. APPLICATIONS

Ce travail est motivé par la résolution d'un certain nombre de problèmes dont les principaux sont présentés ci-dessous.

6.1. Programme linéaire général

La résolution de (P) défini dans le paragraphe 2.

$$\left.\begin{aligned} Ax &= b \\ Bx &\leqq a \\ cx &= z\,(\max) \end{aligned}\right\} \qquad (P)$$

dans l'espace des variables x elles-mêmes (sans être obligé de se ramener dans $\mathbb{R}^n_+$ par translation ce qui n'est possible que si pour tout $i = 1, \ldots, n$, $\operatorname{Min}\{x_i : x \in K\}$ existe ni de décomposer $x = x^+ - x^-$, $x^+ \geqq 0$, $x^- \geqq 0$ pour devoir travailler dans $\mathbb{R}^{2n}_+$, ce qui augmente maladroitement la complexité du problème, d'autant plus que l'application $x \to (x^+, x^-)$ n'échange pas les sommets entre eux) peut se faire grâce à l'algorithme SGGP dans les deux cas suivants :

(*a*) On connaît un sommet x^0 de K.

(*b*) On connaît un point x^0 de K, $x^0 \notin V(K)$.

Dans ce cas on peut démarrer avec la méthode du gradient projeté et obtenir :

* Soit une solution de (P) en x^k sans rencontrer auparavant aucun sommet de K [*i.e.* $x^l \notin V(K)$, $\forall\, l \leqq k-1$].

* Soit un sommet x^k de K et puis continuer la méthode du gradient projeté qui coïncide avec l'algorithme SGGP à partir de cet x^k.

Les situations décrites ci-dessus figurent dans les méthodes d'optimisation globales (pour la résolution des problèmes d'optimisation non convexe) utilisant le principe d'approximation extérieure et la technique des coupes planes [1, 7, 8, 10, 12, 16].

6.2. Détermination de l'ensemble des sommets $V(S^k)$ d'une suite emboîtée de polyèdres convexes bornés

L'élaboration des algorithme d'optimisation globale pose le problème de calcul de $V(S^k)$ où (S^k) est une suite emboîtée de polyèdres convexes bornés (non nécessairement contenus dans $\mathbb{R}^n_+$) qui représente les approximations extérieures : S^{k+1} est défini à partir de S^k en y ajoutant une contrainte linéaire supplémentaire :

$$S^{k+1} = \{ x \in S^k : a_k x \leqq \alpha_k \}.$$

La connaissance de $V(S^k)$ permet de déterminer $V(S^{k+1})$ de la manière suivante [1, 7, 8, 10, 12, 16] : $v \in V(S^{k+1})$ si et seulement si

(*a*) Soit $v \in V(S^k)$ et $a_k v \leqq \alpha_k$.

(*b*) Soit

$$v \in [v_1, v_2] \cap H_k \qquad \text{où} \quad H_k = \{ x : c_k x = a_k \},$$

$v_1 \in V^+(S^k) = \{ x \in V(S^k) : a_k x > \alpha_k \}$ et $v_2 \in V^-(S^k) = \{ x \in V(S^k) : a_k x < \alpha_k \}$.

La détermination d'un sommet $v \in V(S^{k+1})$ du cas (*b*) passe par la détermination (au cas de non dégénérescence) de n sommets adjacents de v_1 (ou de v_2) déjà calculé qui peut être faite par l'algorithme SGGP (*voir* § 3.1.).

6.3. Problème linéaire général dual réalisable

Reprenons le programme linéaire général sous forme canonique (P) défini précédemment :

$$\left.\begin{array}{c} Ax=b \\ Bx\leqq a \\ cx=z\,(\max) \end{array}\right\} \qquad (P)$$

Son problème dual (D) s'écrit

$$\left.\begin{array}{c} yA+vB=c, v\geqq 0 \\ yb+va=w\,(\min). \end{array}\right\} \qquad (DA)$$

On dit que (P) est dual réalisable si l'on dispose d'un ensemble $I\subset\{1, \ldots, p\}$, $|I|=n-m$, tel que :

(i) La matrice $\begin{bmatrix} A \\ B_I \end{bmatrix}$ soit non singulière.

(ii) La solution (λ, μ) du système linéaire :

$$\left.\begin{array}{c} (\lambda, \mu)\begin{bmatrix} A \\ B \end{bmatrix}=c \\ \mu^{I^*}=0 \end{array}\right\} \qquad (14)$$

vérifie $\mu^I\geqq 0$.

Cette définition généralise celle plus connue dans les cas usuels où $K=\{x\in\mathbb{R}^n : Ax=b,\ x\geqq 0\}$ ou $K=\{x\in\mathbb{R}^n : Ax\leqq b,\ x\geqq 0\}$ [4, 13].

Lorsque (P) est dual réalisable, la solution (λ, μ) du (14) est un sommet du polyèdre convexe de solutions réalisables de (D). L'application de l'algorithme SGGP du problème (D) en partant de cette solution (λ, μ) permet de résoudre simultanément les problèmes duaux (P) et (D).

6.4. Programmes linéaires avec variables bornées. Programmes linéaires avec contraintes linéaires additionnelles en post optimisation

Les bornes inférieures et supérieures des variables dans un programme linéaire général peuvent être traitées de différentes façons :

1. Les considérer comme des contraintes « ordinaires » dans un programme linéaire général que l'on résout par SGGP.

2. Les considérer comme des contraintes linéaires additionnelles en post optimisation : On résout en premier lieu le programme linéaire sans tenir compte des contraintes de bornes. Si la solution obtenue vérifie les bornes, elle est une solution optimale. Dans le cas contraire on y ajoute une à une (à commencer par une contrainte de borne la plus violée) ou toutes en même temps pour obtenir des programmes linéaires que l'on résout par SGGP. L'utilisation des contraintes linéaires additionnelles en post optimisation est particulièrement efficace dans :

(i) Les programmes linéaires de formes standard et canonique auxquels des contraintes de bornes sont ajoutées [1, 7, 8, 10, 12-14, 16].

(ii) Optimisation convexe : dans les méthodes d'approximation extérieure dans lesquelles on approxime les fonctions objectives et on linéarise les contraintes [1, 4, 6, 7, 8, 10, 12-14, 16].

(iii) L'optimisation combinatoire et l'optimisation non convexe où l'usage des contraintes linéaires additionnelles en post optimisation est essentiel [1, 7, 8, 10, 12, 14, 16].

CONCLUSION

De par leur démarche originale (une synthèse entre le point de vue d'optimisation convexe proprement dite et celui de l'optimisation combinatoire) les résultats contenus dans ce papier constituent une intéressante contribution à la théorie de programmation linéaire. Ils apportent un élargissement de l'éventail des techniques appropriées de résolution des programmes linéaires.

En dehors des applications spécifiques cités dans le paragraphe 6, la mise en valeur de ces résultats se ferait certainement lors des traitements numériques, par des utilisateurs, des problèmes qui devraient recourir aux techniques les mieux adaptées de la programmation linéaire.

Ce travail s'inscrit dans le cadre de recherches sur l'optimisation globale (optimisation non convexe et non différentiable) menées depuis quelques années au sein de l'Équipe Modélisation et Optimisation de l'Université de Grenoble.

Nous remercions vivement le Referee dont les suggestions ont permis l'amélioration de la présentation de cet article.

BIBLIOGRAPHIE

1. J. CHAARANI, Étude d'une classe d'algorithmes d'optimisation non convexe. Implémentation et Applications, *Thèse de Doctorat*, Université Joseph Fourier, Grenoble, 1989.
2. N. GASTINEL, Analyse Numérique Linéaire, *Hermann*, Paris, 1966.
3. A. KERAGHEL, Étude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar, *Thèse de Doctorat*, Université Joseph Fourier, Grenoble, 1989.
4. D. G. LUENBERGER, Introduction to linear and nonlinear programming, *Addison-Welsey*, 1972.
5. V. H. NGUYEN et J. J. STRODIOT, Computing a global optimal solution to a design centering problem, *Mathematical Programming*, 1990 (to appear).
6. B. PCHENITCHNY, et Y. DANILINE, Méthodes Numériques dans les problèmes d'extrémum, Mir, 1977.
7. PHAM DINH TAO et EL BERNOUSSI SOUAD, Numerical algorithms for solving a class of global nonconvex optimization problems. International Series of Numerical Mathematics: "New Algorithms in Optimization and Their Industrial Use", *Birkhauser Verlag*, 1989.
8. PHAM DINH TAO, JAMAL CHAARANI et EL BERNOUSSI SOUAD, Global numerical algorithms for solving a class of nonconvex optimization problems, *Zeitschrift für Operations Research (ZOR)*, 1989 (to appear).
9. PHAM DINH TAO, Un algortihme pour la résolution du programme linéaire général. Applications. Rapport de Recherches. Université Joseph Fourier, Grenoble, Institut I.M.A.G., 1989.
10. Rapport final du contrat de prestation de recherche Elf-France, 1989 : Conception et Réalisation d'un logiciel d'optimisation non convexe pour la résolution d'un problème d'optimisation de Pool Carburant, Équipe Modélisation et Optimisation, Institut I.M.A.G., Grenoble.
11. R. T. ROCKAFELLAR, Convex analysis, Princeton Univ. Press, 1970.
12. J. B. ROSEN et P. M. PARDALOS, Methods for global concave minimization: a bibliographic survey, *S.I.A.M. Review*, 1988, 28, n° 3.
13. M. SAKAROVITCH, Programmation linéaire, *Hermann*, 1984.
14. M. SAKAROVITCH, Optimisation combinatoire (Méthodes mathématiques et algorithmiques), Programmation discrète, *Hermann*, 1984.
15. A. SCHRIJVER, Theory of linear and integer programming, *John Wiley and Sons*, 1986.
16. A. YASSINE, Études adaptatives et comparatives de certains algorithmes en Optimisation. Implémentations effectives et applications, *Thèse de Doctorat*, Université Joseph Fourier, Grenoble, 1989.