

RAIRO Operations Research

RAIRO Oper. Res. **38** (2004) 63-83

DOI: 10.1051/ro:2004012

**UNE PROCÉDURE DE PURIFICATION
POUR LES PROBLÈMES DE COMPLÉMENTARITÉ
LINÉAIRE, MONOTONES**

ABDERRAHIM KADIRI¹ ET ADNAN YASSINE¹

Communiqué par Jean-Pierre Crouzeix

Abstract. In this paper, we propose a new purification method for monotone linear complementarity problems. This method associates to each iterate of the sequence, generated by an interior point method, one basis which is not necessarily feasible. We show that, under the strict complementarity and non-degeneracy hypotheses, the sequence of bases converges on a finite number of iterations to an optimal basis which gives the exact solution of the problem. The adopted process also serves to preconditioning the conjugate gradient algorithm when computing the Newton direction.

Résumé. Dans cet article, nous proposons une nouvelle méthode de purification pour les problèmes de complémentarité linéaire, monotones. Cette méthode associe à chaque itéré de la suite, générée par une méthode de points intérieurs, une base non nécessairement réalisable. Nous montrons que, sous les hypothèses de complémentarité stricte et de non dégénérescence, la suite des bases converge en un nombre fini d'itérations vers une base optimale qui donne une solution exacte du problème. Le procédé adopté sert également à préconditionner l'algorithme de gradient conjugué lors du calcul de la direction de Newton.

Mots Clés. Problèmes de complémentarité linéaire, méthodes de points intérieurs, purification, solution exacte, convergence finie, gradient conjugué, préconditionnement, programmation quadratique convexe.

Classification Mathématique. 65K05, 90C33, 90C51.

Reçu le 2 septembre 2002.

¹ LMAH, Université du Havre, BP 540, 76058 Le Havre, France ;
e-mail : adnan.yassine@univ-lehavre.fr

© EDP Sciences 2004

INTRODUCTION

Nous considérons le problème de complémentarité linéaire :

$$(LCP) \begin{cases} s &= Mx + q \\ x^T s &= 0 \\ (x, s) &\geq 0 \end{cases}$$

où M est une matrice de type $n \times n$ et q un vecteur de \mathbb{R}^n .

Plusieurs auteurs se sont intéressés durant ces dernières années à la résolution du problème (LCP) par les méthodes de points intérieurs. Ces dernières sont réputées par leur convergence polynomiale, leur rapidité et efficacité et se sont révélées comme de véritables concurrentes des méthodes classiques (simplexe, pivotage de Lemke, etc.). La littérature de ces méthodes a connu une grande expansion et s'est enrichie de plusieurs classes et variantes dans le but de réduire la complexité et améliorer la convergence et l'efficacité. Le lecteur peut voir, par exemple, le livre de Ye [20] qui retrace l'évolution des méthodes de points intérieurs. Cependant, la plupart de ces méthodes ont un inconvénient majeur : la convergence est lente au cours de la phase finale et une solution exacte n'est jamais atteinte.

Pour pallier ce défaut, certains auteurs ont recouru à des procédures de purification. Leur but est d'obtenir une solution exacte de (LCP) en temps polynomial en partant d'une solution approchée, calculée par une méthode de points intérieurs. Une telle procédure a été introduite par Kortanek et Zhu [13] et Ye [19] pour la programmation linéaire. Kojima, Mizuno et Yoshise [11] ont décrit une autre procédure pour les problèmes (LCP) monotones. D'autres méthodes ont été proposées durant cette dernière décennie. Nous citons, par exemple, celles de Monteiro et Wright [16], Ji et Potra [7] et, plus récemment, celles de Potra et Sheng [18], Illés et coll. [6] et Facchinei et coll. [4] pour une classe plus générale, incluant les problèmes (LCP) monotones, où la matrice M est de type P^* [3].

Notre but est de proposer une nouvelle méthode de purification pour le problème (LCP) monotone. Cette méthode combine un algorithme de points intérieurs et un procédé proche de l'algorithme de pivotage de Lemke. C'est une extension des méthodes [11, 13, 19], qui a l'avantage de démarrer à partir de n'importe quel point réalisable (et non une solution approchée) pour obtenir une solution exacte. Nous construisons, parallèlement à la suite de points intérieurs, une autre suite de sommets non nécessairement réalisables qui converge, sous les hypothèses de complémentarité stricte et de non dégénérescence, en un temps fini et polynomial, vers une solution exacte de (LCP). Cette procédure permet aussi de préconditionner l'algorithme du gradient conjugué utilisé pour le calcul de la direction de déplacement. Ce préconditionnement est de plus en plus efficace lorsqu'on s'approche d'une solution optimale.

Nous adaptons cette méthode de purification au programme quadratique convexe après sa transformation en (LCP).

Dans la deuxième section, nous rappelons quelques résultats de purification classiques. Dans la section 3 nous décrivons la nouvelle procédure de purification pour (LCP) et nous montrons la convergence des bases construites à l'aide de cette procédure vers une base optimale qui permet d'identifier une solution optimale exacte. Nous calculons la direction de déplacement en introduisant un préconditionnement de l'algorithme du gradient conjugué et nous énonçons l'algorithme obtenu en combinant une méthode de points intérieurs et la procédure de purification. Dans la section 4 nous adaptons cette procédure au programme quadratique convexe. La section 5 est consacrée aux résultats numériques qui montrent la robustesse et l'efficacité de notre algorithme.

1. PRÉLIMINAIRES

1.1. RAPPELS ET DÉFINITIONS

Définition 1.1. L'ensemble réalisable et son intérieur pour (LCP) sont définis par :

$$S = \{(x, s) \in \mathbb{R}^{2n} : s = Mx + q, x \geq 0, s \geq 0\},$$

$$S_{\text{int}} = \{(x, s) \in S : x > 0, s > 0\}.$$

Dans tout ce qui suit, nous supposons que les hypothèses suivantes sont satisfaites :

- (H1) M est semi-définie positive;
- (H2) $S_{\text{int}} \neq \emptyset$;
- (H3) (LCP) admet une solution (x, s) strictement complémentaire, c'est-à-dire $x_i + s_i > 0 \quad i = 1, 2, \dots, n$.

L'ensemble S des solutions réalisables peut être représenté par le système :

$$Az = q, \quad z \geq 0,$$

où $z = \begin{pmatrix} s \\ x \end{pmatrix}$, $A = [I_n \quad -M]$. I_n étant la matrice identité d'ordre n .

Définition 1.2. On appelle support d'un vecteur $x \in \mathbb{R}^n$ l'ensemble d'indices

$$\sigma(x) = \{i = 1, 2, \dots, n : x_i > 0\}.$$

D'après cette définition, il est facile de vérifier que :

- si $z \in S_{\text{int}}$ alors $\sigma(z) = \Pi = \{1, \dots, 2n\}$;
- si z est un sommet de S alors $\sigma(z) \subset J(z)$, où $J(z)$ désigne l'ensemble d'indices de base de z ;
- $z^* = (x^*, s^*)$ est une solution strictement complémentaire si $\sigma(x^*) \cap \sigma(s^*) = \emptyset$ et $\sigma(x^*) \cup \sigma(s^*) = \{1, 2, \dots, n\}$.

Proposition 1.3 ([19]). *Toutes les solutions optimales strictement complémentaires ont le même support. Autrement dit : Si z^* et \bar{z} sont deux solutions optimales strictement complémentaires alors : $\sigma(z^*) = \sigma(\bar{z})$.*

Nous désignons par C_S la classe des solutions optimales strictement complémentaires, σ^* le support de C_S et $\bar{\sigma}^* = \Pi \setminus \sigma^*$ le complémentaire de σ^* dans Π .

Définition 1.4. L'ensemble des solutions optimales constitue une face du polyèdre des contraintes appelée face optimale. Elle est donnée par :

$$\mathcal{F} = \{z \in S : z_j = 0, \quad \forall j \in \bar{\sigma}^*\}.$$

La taille L du problème (LCP) est définie par [11] :

$$L = E \left(\sum_{i=1}^n \sum_{j=1}^{n+1} \log(|\bar{M}_{ij}| + 1) + \log(n^2) \right) + 1,$$

où $\bar{M} = [M \ q]$ et $E(\xi)$ est la partie entière de ξ .

En fait L est le nombre de bits utilisés dans le codage des données (M, q) supposées entières ; c'est la taille de la mémoire nécessaire pour stocker (M, q) . Le nombre 2^L est très grand, il est supérieur à tous les déterminants mineurs de la matrice des contraintes A d'un problème avec des contraintes linéaires de la forme $\{Ax = b, x \geq 0\}$. En particulier, $2^L/n^2$ est plus grand que tous les déterminants mineurs de $[A \ q]$ pour le problème (LCP) et par conséquent 2^{-L} est un nombre très petit (voir [5] et [11]).

1.2. QUELQUES RÉSULTATS CLASSIQUES DE PURIFICATION

Plusieurs auteurs ont étudié l'évaluation du nombre d'opérations pour déterminer une solution exacte à partir d'une solution approchée pour la programmation linéaire (Kortanek et Zhu [13], Gonzaga [5], Ye [19]). Leurs travaux ont été généralisés pour la programmation quadratique (Monteiro et Adler [15] et Papadimitriou et Steiglitz [17]) et les problèmes de complémentarité linéaire monotones (Kojima, Mizuno et Yoshise [11]). Ces derniers ont montré comment déterminer une solution exacte de (LCP), en partant d'une solution approchée (\hat{x}, \hat{s}) vérifiant

$$\hat{x}_i \hat{s}_i \leq 2^{-2L} \quad i = 1, \dots, n \tag{1}$$

à l'aide d'une procédure qui demande $O(n^3)$ opérations arithmétiques.

Lemme 1.5 ([11]). *Soit $v = (x, s)$ un sommet de S , alors $v_i = 0$ si $v_i < n^2 2^{-L}$.*

Lemme 1.6 ([11]). *Soit $\bar{z} \in S$ et $\bar{K} = \{i : \bar{z}_i < 2^{-L}\}$. Alors il existe un sommet z^* de S tel que $z_i^* = 0$ pour tout $i \in \bar{K}$.*

Théorème 1.7 ([11]). *Soit $\hat{z} = (\hat{x}, \hat{s}) \in S$ vérifiant la condition (1), alors on peut déterminer, en $O(n^3)$ opérations au plus, une solution exacte $z^* = (x^*, s^*)$ de (LCP) telle que*

$$x_i^* = 0 \quad \forall i \in I, \quad \text{et} \quad s_i^* = 0 \quad \forall i \in J,$$

où $I = \{i : \hat{x}_i < 2^{-L}\}$ et $J = \{i : \hat{s}_i < 2^{-L}\}$.

Les résultats précédents montrent l'utilité de la grandeur L et le rôle qu'elle joue pour déterminer une solution exacte et la complexité des algorithmes de points intérieurs à données entières. En outre, ils justifient le choix du test d'arrêt $x^T s \leq 2^{-2L}$ pour le problème (LCP). Cependant, il est difficile dans la pratique d'atteindre la précision 2^{-2L} car ce nombre est très petit, surtout pour des problèmes de grande dimension et à données non creuses. Par conséquent, les procédés de purification cités ci-dessus sont difficiles à appliquer dans la pratique.

Nous citons aussi la propriété suivante qui caractérise les problèmes de type (LCP) [21] :

Proposition 1.8. *Sous l'hypothèse (H3), il existe une partition (B, N) de l'ensemble $\{1, 2, \dots, n\}$ avec $B \cap N = \emptyset$ et $B \cup N = \{1, 2, \dots, n\}$ telle que*

- (1) *si (x^*, s^*) est une solution optimale de (LCP), alors $x_N^* = 0$ et $s_B^* = 0$;*
- (2) *il existe au moins une solution optimale (\bar{x}, \bar{s}) de (LCP) telle que $\bar{x}_B > 0$ et $\bar{s}_N > 0$.*

Le résultat précédent a été utilisé par certains auteurs (Gonzaga et Bonnans [2], Ye et Anstreicher [21]) en vue de caractériser la face optimale. Toutefois, cela reste dans le cadre théorique car, dans la pratique, les algorithmes de points intérieurs ne se basent jamais sur la connaissance de la partition optimale qui est inconnue et se contentent d'une solution approchée (voir [2]). C'est le cas du théorème 1.7 où une partition optimale est identifiée mais à partir de $\hat{z} = (\hat{x}, \hat{s})$ qui est supposée être une solution approchée.

2. MÉTHODE DE PURIFICATION PAR DÉCOMPOSITION G.P.C

Dans cette section, nous décrivons une nouvelle méthode de purification pour le problème (LCP) qui peut être initialisée par n'importe quel point intérieur du domaine réalisable. Nous construisons une suite d'indices J^k qui converge vers une base optimale J^* permettant ainsi de définir une solution exacte de (LCP) par ses indices de base.

2.1. PRINCIPE DE LA MÉTHODE

En plus des hypothèses (H1, H2) et (H3), nous supposons que l'hypothèse suivante est satisfaite :

(H4) tout point z de S est non dégénéré i.e. $\text{rang}(A^{\sigma(z)}) = n$.

Sous ces hypothèses, il existe des algorithmes de points intérieurs qui génèrent une suite (x^k, s^k) dans S_{int} convergeant vers une solution (strictement complémentaire) de (LCP) . On peut voir les articles de Bonnans et Gonzaga [2] et Ye et Anstreicher [21] pour plus d'informations sur ces algorithmes. Nous en citons, par exemple, ceux de Ye et Anstreicher [21], Kojima *et al.* [12], Ji et coll. [8] et McShane [14].

Soit (x^*, s^*) une solution optimale strictement complémentaire. On a donc

$$(x^*)^T s^* = 0, \quad (2)$$

et pour $i = 1, 2, \dots, n$, on a :

$$\frac{x_i^*}{s_i^*} = 0 \quad \text{ou} \quad \frac{x_i^*}{s_i^*} = +\infty.$$

Pour chaque itération k , nous définissons le vecteur ω^k de \mathbb{R}^{2n} par :

$$\omega_i^k = \left(\frac{s_i^k}{x_i^k} \right)^{1/2} \quad i = 1, 2, \dots, n \quad (3)$$

$$\omega_i^k = \left(\frac{x_i^k}{s_i^k} \right)^{1/2} \quad i = n+1, n+2, \dots, 2n. \quad (4)$$

Soit σ^k l'ensemble défini par : $\sigma^k = \{i \in \Pi : \omega_i^k > 1\}$ et $\bar{\sigma}^k = \Pi \setminus \sigma^k$. Chacune des composantes de ω^{σ^k} converge vers $+\infty$ tandis que celles de $\omega^{\bar{\sigma}^k}$ convergent vers 0.

Nous allons construire une suite $\{J^k\} \subset \Pi$, vérifiant $\text{card}(J^k) = n$, avec A^{J^k} non singulière. Les ensembles J^k correspondent aux composantes de ω^k les plus grandes. Nous montrons ensuite que $\{J^k\}$ converge vers une base optimale J^* .

Posons

$$\Omega = \text{diag}\{\omega_j, j \in \Pi\},$$

$$\Omega^J = \text{diag}\{\omega_j, j \in J\},$$

$$\Omega^{\bar{J}} = \text{diag}\{\omega_j, j \in \bar{J}\},$$

où l'indice j est pris dans son ordre de rangement dans les ensembles J ou $\bar{J} = \Pi \setminus J$.

2.2. DÉTERMINATION DE LA DÉCOMPOSITION

En procédant à la décomposition, nous devons tenir compte des trois points suivants :

- (1) $\text{card}(J) = n$;
- (2) les valeurs de $\omega_j, j \in J$ sont « les plus grandes »;
- (3) la matrice A^J est non singulière.

G.P.C désigne la procédure d'éliminations de Gauss avec la stratégie de pivot partiel sur colonnes. La factorisation G.P.C appliquée à la matrice A de type $(n, 2n)$ et de rang n , détermine deux matrices L, U triangulaires inversibles d'ordre n ,

respectivement inférieure et supérieure, une matrice V de type (n, n) et une matrice carrée P d'ordre $2n$ (matrice de permutation) telles que :

$$LAP = [U \ V].$$

Nous désignons par P une matrice de permutation des colonnes de la matrice A , s_p la permutation définie sur Π correspondant à P , J l'image de Π par s_p et $\bar{J} = \Pi \setminus J$.

Définition 2.1. Soit $\Omega = \text{diag}\{\omega_i\}_{i \in \Pi}$, où ω est un vecteur dont toutes les composantes sont strictement positives. Une partition $[J, \bar{J}]$ de Π est dite Ω -partition si et seulement si elle correspond à la factorisation de la matrice $A\Omega$ par la procédure G.P.C :

$$L(A\Omega)P = [U \ V]$$

avec $U = LA^J\Omega^J$, $V = LA^{\bar{J}}\Omega^{\bar{J}}$.

La définition précédente montre que la décomposition dépend à chaque itération k de l'itéré (x^k, s^k) . Nous appliquons cette décomposition à la matrice $A\Omega^k$ où $\Omega^k = \text{diag}\{\omega_i^k\}_{i \in \Pi}$ et ω^k est le vecteur défini en (3) et (4).

Plus (x^k, s^k) s'approche d'une solution optimale, plus des termes ω_j^k deviennent relativement grands et correspondent à des colonnes linéairement indépendantes de A . Ces termes définissent au fur et à mesure les indices d'une base optimale.

Remarque 2.2. Vu que L , U et Ω^J sont inversibles alors la matrice A^J est inversible.

2.3. CONVERGENCE

Définition 2.3. Soit u un vecteur de \mathbb{R}^{2n} et $r \in \{1, \dots, n\}$ tel que $u_r \neq 0$. On définit la matrice d'élimination notée $E(u, r)$ ou en abrégé E par :

$$Eh = d \quad \text{avec} \quad d_i = \begin{cases} h_i & \text{pour } i = 1, 2, \dots, r, \\ h_i - \left(\frac{u_i}{u_r}\right)h_r & \text{pour } i = r + 1, \dots, n, \end{cases}$$

pour tout $h \in \mathbb{R}^n$.

Lemme 2.4. Si α est un réel non nul, u un vecteur dans \mathbb{R}^n et r un indice vérifiant $1 \leq r \leq n$ et $u_r \neq 0$, alors on a $E(\alpha u, r) = E(u, r)$.

Démonstration. Conséquence directe de la définition de la matrice d'élimination. \square

Si $\Omega^k = \text{diag}\{\omega_i^k\}_{i \in \Pi}$ et ω^k est le vecteur défini en (3) et (4) on a :

$$(A\Omega^k)_{i,j} = A_{i,j}\omega_j^k \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, 2n.$$

En désignant respectivement par A^j et $(A\Omega^k)^j$ le vecteur colonne de A et de $A\Omega^k$ d'indice j , on obtient donc

$$(A\Omega^k)^j = A^j\omega_j^k \quad j = 1, 2, \dots, 2n.$$

On peut alors vérifier facilement grâce au lemme 2.4 que les matrices d'élimination qui interviennent dans la procédure G.P.C ne dépendent pas de Ω^k .

Lemme 2.5. *Si G est une matrice inversible d'ordre $p \leq n$, il existe alors une permutation t de l'ensemble $\{1, 2, \dots, p\}$ vérifiant :*

$$|G_{t(i),i}| > 0 \quad \text{pour tout } i \in \{1, 2, \dots, p\}.$$

Démonstration. Le déterminant de G est donné par l'expression :

$$\det(G) = \sum_{t \in T_p} \varepsilon_t G_{t(1),1} G_{t(2),2} \dots G_{t(p),p}$$

T_p étant l'ensemble des permutations de $\{1, 2, \dots, p\}$ et ε_t la signature de t . Puisque G est non singulière, son déterminant est non nul, nous avons donc

$$\sum_{t \in T_p} |\varepsilon_t G_{t(1),1} G_{t(2),2} \dots G_{t(p),p}| > 0.$$

Par conséquent il existe $\bar{t} \in T_p$ tel que pour tout $i \in \{1, 2, \dots, p\}$, $|G_{\bar{t}(i),i}| > 0$. \square

Le résultat de convergence finie de la suite $\{J^k\}$ vers J^* base optimale est donné par le théorème suivant :

Théorème 2.6. *Soit $[J^k, \bar{J}^k]$ une Ω^k -partition de Π où ω^k le vecteur défini en (3) et (4) et $\Omega^k = \text{diag}\{\omega_i^k\}_{i \in \Pi}$. Il existe un rang K à partir duquel J^k est une base optimale.*

Démonstration. La face optimale $\mathcal{F} = \{z \in S : z_j = 0, \quad \forall j \in \bar{\sigma}^*\}$ contient $z^* = (s^*, x^*)$ limite de la suite (s^k, x^k) . D'après l'hypothèse (H3) il est facile de vérifier que $\text{card}(\sigma^*) = n$. D'autre part, l'hypothèse (H4) assure que $\text{rang}(A^{\sigma^*}) = n$ i.e. A^{σ^*} est inversible et que \mathcal{F} est réduite au point z^* (sommet optimal), donné par : $z_{\sigma^*}^* = (A^{\sigma^*})^{-1}q$ et $z_{\bar{\sigma}^*}^* = 0$, où $z_{\sigma^*}^*$ (resp. $z_{\bar{\sigma}^*}^*$) est le vecteur formé par les composantes de z^* dont les indices appartiennent à σ^* (resp. $\bar{\sigma}^*$).

Remarquons que si J est une partie de Π vérifiant $\sigma^* \subset J$ et $\text{card}(J) = n$ alors $J = \sigma^*$ est la base optimale correspondant au sommet z^* . Pour montrer que la suite $\{J^k\}$ converge vers la base optimale, il suffit donc de montrer qu'il existe K tel que $\sigma^* \subset J^k \quad \forall k \geq K$.

Soit $i \in \sigma^*$ i.e. $z_i^* > 0$.

- i) Si $1 \leq i \leq n$ alors $z_i^* = s_i^* > 0$ et $x_i^* = 0$. D'où $\omega_i^k = \left(\frac{s_i^k}{x_i^k}\right)^{1/2} \rightarrow \frac{s_i^*}{x_i^*}^{1/2} = +\infty$.
- ii) Si $n+1 \leq i \leq 2n$ alors $z_i^* = x_i^* > 0$ et $s_i^* = 0$ et par suite $\omega_i^k = \left(\frac{x_i^k}{s_i^k}\right)^{1/2} \rightarrow \frac{x_i^*}{s_i^*}^{1/2} = +\infty$.

On a donc $\omega_i^k \rightarrow +\infty \quad \forall i \in \sigma^*$.

Soit maintenant $j \in \bar{\sigma}^*$, donc $z_j^* = 0$.

i) Si $1 \leq j \leq n$ alors $z_j^* = s_j^* = 0$ et $x_i^* > 0$ (d'après l'hypothèse (H3)) et

$$\text{par suite } \omega_j^k = \left(\frac{s_j^k}{x_j^k} \right)^{1/2} \rightarrow \frac{s_j^*}{x_j^*}^{1/2} = 0.$$

ii) Si $n+1 \leq j \leq 2n$ alors $z_j^* = x_j^* = 0$ et $s_i^* > 0$ (d'après (H3)), donc

$$\omega_j^k = \left(\frac{x_j^k}{s_j^k} \right)^{1/2} \rightarrow \frac{x_j^*}{s_j^*}^{1/2} = 0.$$

Donc $\omega_j^k \rightarrow 0 \forall j \in \bar{\sigma}^*$.

On déduit de ce qui précède et du lemme 2.4 que pour tout $j \in \bar{\sigma}^*$ on a

$$\forall \varepsilon > 0 \quad \exists K_1 \in \mathbb{N} : \quad \forall k \geq K_1 \quad \| (L^k A \Omega^k)^j \| \leq \varepsilon$$

où L^k est un produit de matrices d'éliminations.

En tenant compte du lemme 2.5, il existe une permutation t dans T_n vérifiant

$$| A_{t(i),i}^{\sigma^*} | > \mu \quad \forall i = 1, \dots, n$$

où μ est un réel strictement positif.

Nous avons $A_{t(i),i}^{\sigma^*} = A_{t(i),j_i}$, où j_i est le i ème indice dans σ^* ($\sigma^* = \{j_1, j_2 \dots j_n\}$).

Or la matrice $G^k = (L^k A \Omega^k)^{\sigma^*}$ a le même rang que A^{σ^*} i.e. $\text{rang}(G^k) = n$, donc nous avons

$$| G_{t^k(i),i}^k | > 0 \quad \forall i = 1, \dots, n$$

où $t^k \in T_n$.

Pour tout $i \in \{1, \dots, n\}$ nous avons $G_{t^k(i),i}^k = (L^k A \Omega^k)_{t^k(i),j_i}$. Comme $\omega_j^k \rightarrow +\infty$ pour tout $j \in \sigma^*$ et tenant compte du lemme 2.4, on déduit que $| G_{t^k(i),i}^k | \rightarrow +\infty$, d'où

$$\forall B > 0 \quad \exists K_2 \in \mathbb{N} : \quad \forall k \geq K_2 \quad | G_{t^k(i),i}^k | > B.$$

Nous pouvons toujours choisir $\varepsilon > 0$ vérifiant :

$$\varepsilon < B \quad \text{et} \quad \| (L^k A \Omega^k)^j \| < \varepsilon \quad \forall j \in \bar{\sigma}^* \quad \forall k \geq K = \sup(K_1, K_2).$$

Puisque le pivot doit être choisi maximal en valeur absolue, il existe n pivots qui ne peuvent être choisis par la procédure G.P.C que parmi les colonnes de A^{σ^*} . Autrement dit $\sigma^* \subset J^k \quad \forall k \geq K$ et par suite $\sigma^* = J^k \quad \forall k \geq K$. \square

Corollaire 2.7. *Sous les hypothèses (H1, H2, H3) et (H4), l'algorithme de points intérieurs couplé avec la procédure de décomposition donne en un nombre fini d'itérations une base optimale J^* .*

Démonstration. Il suffit de prendre $J^* = J^K$. \square

Remarque 2.8. (J^*, \bar{J}^*) est la partition optimale de $\{1, \dots, 2n\}$ qui correspond à la solution optimale z^* et qui vérifie la propriété 1.8.

2.4. CALCUL DE LA DIRECTION DE DÉPLACEMENT
PAR LA DÉCOMPOSITION G.P.C

Dans ce paragraphe, nous exposons une méthode pour calculer la direction de déplacement en utilisant la décomposition G.P.C.

En général, la direction de Newton (d_x, d_s) est donnée par le système

$$\begin{cases} Sd_x + Xd_s = -XSe + \gamma\mu e, \\ d_s = Md_x, \end{cases} \quad (5)$$

où $X = \text{diag}(x)$, $S = \text{diag}(s)$, $\mu = x^T s/n$ et $0 < \gamma < 1$.

En multipliant la première équation par la matrice $X^{-1/2}S^{-1/2}$, on obtient

$$\begin{cases} X^{-1/2}S^{1/2}d_x + X^{1/2}S^{-1/2}d_s = X^{-1/2}S^{-1/2}(-XSe + \gamma\mu e), \\ d_s = Md_x. \end{cases} \quad (6)$$

Posons $D = X^{1/2}S^{-1/2}$. La première équation suggère de considérer le changement de variable

$$\bar{d}_x = D^{-1}d_x, \quad \bar{d}_s = Dd_s,$$

qui correspond à une mise à l'échelle (*scaling*) du système (5). Nous obtenons

$$\begin{cases} \bar{d}_x + \bar{d}_s = \bar{v}, \\ MD\bar{d}_x - D^{-1}\bar{d}_s = 0, \end{cases} \quad (7)$$

où $\bar{v} = X^{-1/2}S^{-1/2}(-XSe + \gamma\mu e)$.

La deuxième équation du système précédent peut s'écrire

$$[I_n \quad -M] \begin{bmatrix} D^{-1} & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} \bar{d}_s \\ \bar{d}_x \end{bmatrix} = 0. \quad (8)$$

Or $\begin{bmatrix} D^{-1} & 0 \\ 0 & D \end{bmatrix} = \Omega = \text{diag}\{\omega_i\}_{i \in \Pi}$ et $[I_n \quad -M] = A$, où ω est le vecteur défini en (3) et (4).

Ainsi, la dernière équation équivaut à

$$A\Omega \begin{bmatrix} \bar{d}_s \\ \bar{d}_x \end{bmatrix} = 0.$$

En appliquant la décomposition G.P.C à la matrice $A\Omega$, on obtient l'équation équivalente

$$[U \quad V] \begin{bmatrix} \bar{d}_s \\ \bar{d}_x \end{bmatrix} = 0.$$

Soit

$$U\bar{d}_s + V\bar{d}_x = 0,$$

ou d'une manière équivalente

$$\bar{d}_s = -U^{-1}V\bar{d}_x.$$

Par suite (\bar{d}_x, \bar{d}_s) est donnée par le système d'équations

$$\begin{cases} \bar{d}_s + \bar{d}_x = \bar{v}, \\ U\bar{d}_s + V\bar{d}_x = 0. \end{cases} \quad (9)$$

En conclusion, nous calculons \bar{d}_x en résolvant le système linéaire réduit

$$(U - V)\bar{d}_x = U\bar{v}. \quad (10)$$

\bar{d}_s est calculé par :

$$\bar{d}_s = \bar{v} - \bar{d}_x.$$

Ensuite

$$d_x = D\bar{d}_x, \quad d_s = D^{-1}\bar{d}_s.$$

Rappelons que U est une matrice triangulaire supérieure et inversible d'ordre n et V est une matrice d'ordre n . En plus, $(U - V)$ n'est pas symétrique mais inversible. En effet, soit $x \in \mathbb{R}^n$ tel que

$$(U - V)x = 0.$$

On a alors

$$[U \quad V] \begin{bmatrix} x \\ -x \end{bmatrix} = 0.$$

Soit

$$[I_n \quad -M] \begin{bmatrix} D^{-1} & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} x \\ -x \end{bmatrix} = 0,$$

qui nous donne

$$MDx + D^{-1}x = 0,$$

ou encore

$$(M + D^{-2})x = 0.$$

Puisque M est semi-définie positive et D^{-2} est diagonale inversible, alors $(M + D^{-2})$ est inversible. D'où $x = 0$.

En tenant compte de la structure de la matrice $(U - V)$, nous appliquons la méthode de gradient conjugué qui présente l'avantage d'être stable et rapide. En outre, pour contrôler sa stabilité, nous effectuons un préconditionnement au système (10). Le préconditionnement utilisé est décrit au paragraphe suivant.

2.5. PRÉCONDITIONNEMENT DE L'ALGORITHME DU GRADIENT CONJUGUÉ

La direction de déplacement est donnée par le système

$$\begin{cases} (U - V)\bar{d}_x = U\bar{v}, \\ \bar{d}_s = \bar{v} - \bar{d}_x. \end{cases} \quad (11)$$

La première équation est de la forme

$$Hd = u,$$

où, pour des raisons de simplicité, nous avons posé $H = U - V$, $\bar{d}_x = d$ et $U\bar{v} = u$.

Comme $H = U - V$ est inversible non symétrique, nous appliquons l'algorithme de gradient conjugué au système

$$H^T H d = H^T u. \quad (12)$$

Plusieurs approches ont été décrites dans la littérature, pour réduire le conditionnement et améliorer la vitesse de convergence et la performance de l'algorithme du gradient conjugué. Ces approches varient suivant la structure du système linéaire à résoudre. Le principe consiste à remplacer ce dernier par un autre système linéaire symétrique défini positif ayant la même solution et une matrice dont le conditionnement est proche de 1. Parmi ces approches, nous appliquons celle qui s'adapte à notre cas, où la matrice du système s'écrit $H = B^T B + C$, avec B matrice triangulaire inférieure ou supérieure et C une matrice dont les termes sont petits. Cette approche est appelée « preconditionnement par factorisation incomplète » (incomplete factorisation preconditioners) citée dans [10].

En effet, nous avons

$$\begin{aligned} H^T H &= (U - V)^T (U - V) \\ &= U^T U - U^T V - V^T U + V^T V. \end{aligned}$$

D'après la définition 2.1, les matrices U et V sont données par :

$$U = LA^J \Omega^J, \quad V = LA^{\bar{J}} \Omega^{\bar{J}}.$$

Or les termes de la matrice Ω^J convergent vers $+\infty$ tandis que ceux de $\Omega^{\bar{J}}$ convergent vers 0. Donc, plus on s'approche d'une solution optimale plus $\text{cond}(H^T H)$ est proche de $\text{cond}(U^T U)$. Ainsi la matrice $U^T U$ sert comme matrice de conditionnement du système (10).

L'algorithme qui va suivre est déduit de l'algorithme du gradient conjugué en introduisant un préconditionnement du système (10) :

Algorithme 2.9. Gradient conjugué préconditionné

Début

Initialisation : $d^0 \in \mathbb{R}^n$, $r^0 := H^T u - H^T H d^0$, $p^0 := (U^T U)^{-1} r^0$.

Étape 1 : $\mu := \langle r, (U^T U)^{-1} r \rangle / \langle p, H^T H p \rangle$.

Étape 2 : $d' := d + \mu p$.

Étape 3 : $r' := r - \mu H^T H p$.

Étape 4 : $\beta := \langle r', (U^T U)^{-1} r' \rangle / \langle r, (U^T U)^{-1} r \rangle$.

Étape 5 : $p' := r' + \beta p$.

Fin

Test d'arrêt : $\| r \| \leq \epsilon$ où ϵ est un réel positif assez petit.

2.6. MISE EN ŒUVRE DE LA DÉCOMPOSITION G.P.C

Première décomposition : pour décomposer la matrice A à l'étape $k=1$, nous procédons comme suit.

Nous classons les ω_i par ordre décroissant et nous prenons J correspondant à cet ordre : $\omega_{j_1} \geq \omega_{j_2} \geq \dots \geq \omega_{j_n}$, puis nous classons les colonnes A^j , $j \in J$ suivant le degré croissant de densité; c'est-à-dire, suivant cet ordre la première colonne contient plus de zéros que la deuxième, la deuxième contient plus de zéros que la troisième...

Lors des éliminations nous choisissons parmi les composantes de la colonne en cours le pivot maximal en valeur absolue, Si ce pivot est inférieur à une certaine tolérance nous passons à la colonne suivante..., jusqu'à compléter les éliminations de n colonnes.

kème décomposition : avant d'entamer la description de la décomposition à l'itération k , signalons la remarque suivante.

Généralement, le classement des valeurs de ω_j ne varie pas beaucoup d'une itération à une autre, en conséquence, J^k ne diffère pas beaucoup de J^{k-1} .

C'est pour cette raison que nous avons mis à jour une procédure exploitant ce fait, elle est décrite comme suit.

Nous classons les valeurs ω_j , $j = 1, 2, \dots, 2n$ suivant l'ordre décroissant et nous prenons J l'ensemble des n premiers indices correspondant à ce classement, autrement dit, $J = \{j_1, j_2, \dots, j_n\}$ tel que $\omega_{j_1} \geq \omega_{j_2} \geq \dots \geq \omega_{j_n} \geq \omega_{j_{n+1}} \geq \dots \geq \omega_{j_{2n}}$ et $\bar{J} = \{j_{n+1}, j_{n+2}, \dots, j_n\}$. Nous posons $\tilde{J} = J \cap J^{k-1}$ et nous gardons dans le premier bloc (dans la partie des n premières colonnes) les colonnes correspondant à \tilde{J} à un ordre près. Nous obtenons donc, moyennant la remarque faite au début, une partie des colonnes déjà éliminée (contenant des zéros) et nous complétons les éliminations nécessaires qui restent. Nous aurons en finalité la matrice triangulaire, supérieure inversible, d'ordre n , installée dans le premier bloc et notée U . Le choix

de pivot est similaire à celui de la première décomposition (itération $k = 1$). Il est possible aussi de choisir le pivot sur colonnes au lieu des lignes.

Nous résumons ce qui précède par les deux procédures suivantes :

Procédure G.P.C.1

Début (entrées : ω, A)

Étape 1 : Classer les colonnes de A suivant le degré croissant de densité (réordonner A).

Étape 2 : Classer les ω_j suivant l'ordre décroissant.

Étape 3 : Pour la colonne A^1 , choisir $p = \max\{|A_{i1}|, i = 1, 2, \dots, n\}$ comme pivot.

Étape 4 : Éliminer les " $n - 1$ " valeurs de A^1 .

Étape 5 : Pour $j \geq 2$ et (*stop=faux*) faire : $p = \max\{|A_{ij}|, i = 1, 2, \dots, n\}$.

Si $p \leq \varepsilon$ alors passer à $j + 1$ et recommencer le test jusqu'à trouver $p > \varepsilon$.

stop=vrai dès que les éliminations sont portées sur n colonnes.

Fin (Sortie : J, \bar{J}, U, V)

Procédure G.P.C.k

Début (entrées : ω, A, J)

Étape 1 : Classer les ω_j selon l'ordre décroissant, et prendre J' correspondant.

Étape 2 : Prendre $\bar{J} = J \cap J'$.

Étape 3 : Classer les colonnes de A selon l'ordre décroissant des zéros qu'elles contiennent.

Étape 4 : Compléter les éliminations pareillement à l'étape 5 de G.P.C.1.

Fin (sortie : J', \bar{J}', U', V')

Commentaires :

- en procédant ainsi, le coût de G.P.C est incontestablement faible, notamment pour les matrices creuses ;
- l'accumulation des erreurs n'atteint pas un degré gênant, ceci grâce au nombre non élevé des itérations dans les méthodes de point intérieurs en général ;
- la procédure G.P.C décrite ici ne correspond pas exactement à celle décrite théoriquement, néanmoins la convergence de la suite $\{J^k\}$ vers une base optimale J^* est garantie, ceci grâce aux deux points suivants :
 - (1) la façon de classer les ω_j ;
 - (2) la façon de choisir le pivot (donnant A^J inversible).

Test d'optimalité de J^k

Le test d'optimalité est analogue à celui de l'algorithme du pivotage de Lemke, car il s'agit tout simplement de tester l'optimalité d'une base non nécessairement

réalisable. Ce test est donc donné par :

$$(T1) \quad z_J = (A^J)^{-1}q \geq 0;$$

$$(T2) \quad i \in J \iff i + n \in \bar{J} \text{ et } i \in \bar{J} \iff i + n \in J, \quad i = 1, \dots, n.$$

(T1) est un test de réalisabilité. Tandis que (T2) correspond aux conditions de complémentarité $x_i s_i = 0, i = 1, \dots, n$ et signifie que l'une des deux composantes x_i et s_i est une variable de base et l'autre est une variable hors base. Notons que si $z = (x, s)^T$, alors $z_i = x_i$ et $z_{i+n} = s_i$ pour $i = 1, \dots, n$.

Ces deux tests sont très simples à effectuer grâce à la décomposition de la matrice A .

2.7. ALGORITHME PURIFIÉ

Dans ce paragraphe nous énonçons l'algorithme obtenu en combinant la procédure G.P.C avec un algorithme de points intérieurs sous les hypothèses du paragraphe 3.1.

Algorithme 2.10. Points intérieurs avec purification

Données : n, M, q .

Initialisation : calculer (x^0, s^0) .

- 1) Calculer ω .
- 2) Si $k = 0$ faire G.P.C.1 sinon faire G.P.C.k.
- 3) Si J vérifie (T1) et (T2) stop : J est la base optimale et $(z_J, z_{\bar{J}})$ est la solution optimale.
- 4) Calculer (d_x, d_s) en résolvant le système (10) par la procédure 2.9.
- 5) $(x', s') = (x + \theta d_x, s + \theta d_s)$.
- 6) $k \leftarrow k + 1$.

Remarques

- Le point initial (x^0, s^0) peut être calculé par la procédure d'initialisation décrite dans [11].
- Le calcul du pas de déplacement θ varie suivant l'algorithme combiné avec la procédure G.P.C ([8, 12, 14, 21], ...).
- Dans l'algorithme 2.10, la procédure de purification est utilisée à partir de la première itération. Toutefois, elle peut commencer à partir d'une itération k telle que le saut primal-dual $(x^k)^T s^k$ est suffisamment petit afin d'économiser le coût de la décomposition. En effet, l'analyse de convergence des algorithmes cités ci-dessus montre que le taux de réduction de $(x^k)^T s^k$ est très grand aux premières itérations et plus on s'approche d'une solution optimale plus ce taux devient petit. Par exemple, dans l'algorithme de Ji *et al.* qui a une convergence superlinéaire, $(x^k)^T s^k$ est réduit par un facteur qui tend vers 0 (voir par exemple [8], p. 194 et 197).

3. APPLICATION DE LA PURIFICATION G.P.C À LA PROGRAMMATION QUADRATIQUE CONVEXE

3.1. CAS DE CONTRAINTES $Ax \leq b \quad x \geq 0$

Soit le problème quadratique convexe :

$$(P^1) \begin{cases} \min 1/2x^T Qx + c^T x \\ Ax \leq b, \\ x \geq 0, \end{cases}$$

où Q est une matrice carrée d'ordre n symétrique semi-définie positive ; A est une matrice de type $m \times n$; c est un vecteur de \mathbb{R}^n et b un vecteur de \mathbb{R}^m .

Un point x réalisable pour (P^1) est une solution optimale si et seulement s'il existe $y \in \mathbb{R}_+^m$ et $z \in \mathbb{R}_+^n$ vérifiant :

$$\begin{aligned} Qx + c + A^T y - z &= 0, \\ z_i x_i &= 0, \quad i = 1, 2, \dots, n, \\ y_i r_i &= 0, \quad i = 1, 2, \dots, m, \end{aligned}$$

avec $r = b - Ax$.

(P^1) est donc équivalent au problème de complémentarité linéaire suivant :

$$(LCP^1) \begin{cases} w = Mu + q, \\ w^T u = 0, \\ (w, u) \geq 0, \end{cases}$$

avec $w = \begin{pmatrix} z \\ r \end{pmatrix}$, $u = \begin{pmatrix} x \\ y \end{pmatrix}$, $q = \begin{pmatrix} c \\ b \end{pmatrix}$, $M = \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix}$.

On peut facilement vérifier que la matrice M est semi-définie positive.

L'ensemble réalisable et son intérieur pour (LCP^1) s'écrivent :

$$\begin{aligned} S &= \{(u, w) \in \mathbb{R}^{2(n+m)} : w = Mu + q, w \geq 0, u \geq 0\}, \\ S_{int} &= \{(u, w) \in S : u > 0, w > 0\}. \end{aligned}$$

Nous supposons que les hypothèses $(H2, H3)$ et $(H4)$ sont satisfaites.

3.2. CAS DE CONTRAINTES $Ax = b \quad x \geq 0$

Soit le problème quadratique convexe :

$$(P^2) \begin{cases} \min 1/2x^T Qx + c^T x \\ Ax = b, \\ x \geq 0 \end{cases}$$

où Q est une matrice de type $n \times n$ symétrique semi-définie positive, A une matrice $m \times n$, $m \leq n$ de rang m , $c \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

Les conditions d'optimalité nécessaires et suffisantes de (P^2) s'écrivent :

$$\begin{cases} -Qx + A^T w + s = c, \\ Ax = b, \\ x^T s = 0, \\ (x, s) \geq 0. \end{cases} \quad (13)$$

Soit B une matrice de type $(n - m) \times n$ dont les vecteurs lignes engendrent le noyau de A , on a donc $AB^T = 0$.

En multipliant la première équation de (13) par B , on obtient l'équation équivalente $-BQx + Bs = Bc$.

Le système (13) est donc équivalent au système :

$$\begin{cases} \begin{bmatrix} A & 0 \\ -BQ & B \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} b \\ Bc \end{bmatrix}, \\ x^T s = 0, \\ (x, s) \geq 0. \end{cases} \quad (14)$$

Par conséquent, le problème (P^2) est équivalent au problème de complémentarité linéaire :

$$(LCP^2) \begin{cases} Hx + Ts = h, \\ x^T s = 0, \\ x \geq 0, s \geq 0, \end{cases}$$

avec $H = \begin{bmatrix} A \\ -BQ \end{bmatrix}$, $T = \begin{bmatrix} 0 \\ B \end{bmatrix}$ et $h = \begin{bmatrix} b \\ Bc \end{bmatrix}$.

(LCP^2) correspond à la formulation générale d'un problème de complémentarité linéaire. On a montré qu'il peut être ramené à un problème sous forme standard de type (LCP^1) à l'aide de transformations qui consistent à appliquer des permutations sur les lignes ou les colonnes de H et T et les composantes de h (voir [1,2]).

L'ensemble réalisable de (LCP^2) et son intérieur s'écrivent :

$$\begin{aligned} F &= \{(x, s) \in \mathbb{R}^{2n} : Hx + Ts = h, x \geq 0, s \geq 0\}, \\ F^0 &= \{(x, s) \in F : x > 0, s > 0\}. \end{aligned}$$

Les hypothèses $(H1)$, $(H2)$, $(H3)$ et $(H4)$ se formulent pour le problème (LCP^2) de la façon suivante [1,2] :

- $(H1')$ (LCP^2) est monotone *i.e.* pour tout $(u, v) \in \mathbb{R}^{2n}$, $Hu + Tv = 0 \implies u^T v \geq 0$;
- $(H2')$ $F^0 \neq \emptyset$;
- $(H3')$ (LCP^2) admet une solution optimale $z^* = (x^*, s^*)$ telle que $x^* + s^* > 0$;
- $(H4')$ tout point $z = (x, s)$ de F est non dégénéré, *i.e.* $\text{rang}(G^{\sigma(z)}) = n$, où $G = \begin{bmatrix} H & T \end{bmatrix}$.

Comme Q est semi-définie positive (H1) est toujours vérifiée. En effet, soit $u, v \in \mathbb{R}^n$ tel que $Hu + Tv = 0$. Donc, $Au = 0$ et $-BQu + Bv = 0$. Soit : $B(-Qu + v) = 0$. D'où il existe $y \in \mathbb{R}^m$ tel que : $-Qu + v = A^T y$.

En multipliant la dernière équation par u^T , on obtient : $u^T v = u^T Q u \geq 0$.

Nous supposons que les hypothèses (H2', H3') et (H4') sont vérifiées.

En conclusion, un problème quadratique convexe de type (P¹) (resp. (P²)) est équivalent à un problème de complémentarité linéaire monotone. La résolution de ce dernier par un algorithme de point intérieur combiné avec la procédure de purification, sous les hypothèses (H2, H3) et (H4) (resp. (H2'), (H3') et (H4')) aboutit à la solution optimale primale-duale exacte de (P¹) (resp. (P²)).

Remarque 3.1. La matrice B est obtenue par la factorisation QR de la matrice A^T . En effet, cette dernière donne :

$$A^T = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

avec Q matrice carrée orthogonale d'ordre n , R matrice triangulaire supérieure inversible de type $m \times m$, Q_1 et Q_2 de type $n \times m$ et $n \times (n - m)$ respectivement.

Les vecteurs colonnes de Q forment une base orthogonale de \mathbb{R}^n , et on a :

$$Q_1^T Q_2 = Q_2^T Q_1 = 0.$$

D'où $Q_2^T A^T = Q_2^T Q_1 R = 0$. On en déduit que $B = Q_2^T$ vérifie : $BA^T = 0$, soit $AQ_2 = AB^T = 0$ et les vecteurs lignes de B forment une base de $\ker A$.

4. SIMULATIONS NUMÉRIQUES

Ces tests numériques ont été effectués en utilisant comme algorithme de base l'algorithme de Monteiro et Adler [15] qui est très utilisé pour la résolution des programmes quadratiques convexes et qui fait partie des algorithmes de suivi de trajectoire centrale. Ces derniers ont la meilleure complexité connue à ce jour à savoir $O(\sqrt{n}L)$ et une convergence rapide.

Nous transformons un programme quadratique convexe de type P^1 en (LCP). TC désigne l'algorithme de Monteiro et Adler pour la résolution de P^1 et TCP cet algorithme avec purification. Les données sont aléatoires.

dim : dimension de PQ ($n \times m$). Iter : nombre d'itérations.

Les tests ont été réalisés sur une machine AMD Athlon XP 1600+ en langage C.

Commentaires

Nous remarquons que le nombre d'itérations de la méthode TCP est toujours plus faible que celui de la méthode TC et que le temps de calcul global est toujours plus petit avec un taux de réduction proche de 15 p.c. Les simulations numériques effectués sur plusieurs problèmes montrent l'efficacité et la rapidité de notre algorithme.

TABLEAU 1. Comparaison du temps d'exécution et nombre d'itérations.

Problème	Méthode TC		Méthode TCP		
	dim	t(sec)	Iter	t(sec)	Iter
Pb1	30x20	2	12	1	9
Pb2	30x20	2	11	2	7
Pb3	30x20	2	12	1	9
Pb4	60x50	4	13	2	9
Pb5	60x50	4	12	1	9
Pb6	60x50	3	12	1	9
Pb7	100x55	6	14	4	11
Pb8	100x55	5	13	4	10
Pb9	100x55	6	13	3	10
Pb10	130x80	15	16	11	11
Pb11	130x80	13	14	9	11
Pb12	130x80	16	17	10	12
Pb13	200x 150	67	16	49	14
Pb14	200x 150	68	16	51	13
Pb15	200x 150	62	15	45	12
Pb16	350x 300	362	16	252	12
Pb17	350x 300	363	16	249	13
Pb18	350x 300	412	18	259	14
Pb19	350x 300	340	19	249	12
Pb20	350x 300	363	19	250	14
Pb21	550x 400	1103	21	935	15
Pb22	550x 400	1174	19	944	15
Pb23	550x 400	1183	23	930	16
Pb24	600x 500	1394	25	1089	21
Pb25	600x 500	1532	26	1211	20
Pb26	600x 500	1540	26	1250	21

5. CONCLUSION

Dans cet article nous avons proposé une nouvelle méthode de purification pour les problèmes de complémentarité linéaire, monotones. C'est une extension des méthodes décrites dans [11, 13, 15, 19]. Nous avons introduit une procédure qui peut être combinée avec les algorithmes de points intérieurs utilisant les hypothèses de complémentarité stricte et de non dégénérescence. Ces hypothèses ne sont pas restrictives et ont été utilisées par plusieurs auteurs (Ye et Anstreicher [21], Kojima *et al.* [12], Ji *et al.* [8], McShane [14] entre autres) pour généraliser des résultats obtenus pour la programmation linéaire et atteindre une convergence superlinéaire voire quadratique (voir [2] et [21] pour plus de détails sur ces méthodes).

Notre méthode a l'avantage de démarrer à partir de n'importe quel point réalisable pour calculer explicitement la base optimale et obtenir la solution exacte

de (*LCP*). Enfin cette méthode permet un préconditionnement de l'algorithme du gradient conjugué, pour le calcul de la direction de déplacement, qui est de plus en plus efficace au voisinage de l'optimum.

D'autres auteurs ont proposé des méthodes qui sont différentes de la nôtre. Nous citons, par exemple, [7, 16, 18] pour les problèmes (*LCP*) dégénérés et plus récemment [4, 6] pour une classe qui inclue les problèmes (*LCP*) monotones, où la matrice M est de type P_* .

Notre étude peut être étendue aux programmes convexes à contraintes linéaires et aussi aux programmes non linéaires qui se ramènent, *via* des méthodes (SQP), à des sous-problèmes quadratiques convexes.

RÉFÉRENCES

- [1] J.F. Bonnans, J.C. Gilbert, C. Lemarechal and C. Sagastizabal, *Optimisation Numérique. Aspects théoriques et pratiques*. Springer-Verlag (1997).
- [2] J.F. Bonnans and C.C. Gonzaga, Convergence of interior point algorithms for the monotone linear complementarity problem. *Math. Oper. Res.* **21** (1996) 1-25.
- [3] R.W. Cottle, J.S. Pang and V. Venkateswaran, Sufficient matrices and the linear complementarity problem. *Linear Algebra Appl.* **114/115** (1989) 231-249.
- [4] F. Facchinei, A. Fischer and C. Kanzow, On the identification of zero variables in a interior-point framework. *SIAM J. Optim.* **10** (2000) 1058-1078.
- [5] C.C. Gonzaga, Path-following methods for linear programming. *SIAM Rev.* **34** (1992) 167-224.
- [6] T. Illes, J. Peng, C. Roos and T. Terlaky, A strongly polynomial rounding procedure yielding a maximally complementary solution for $P_*(\kappa)$ linear complementarity problems. *SIAM J. Optim.* **11** (2000) 320-340.
- [7] J. Ji and A. Potra, *Tapia indicators and finite termination of infeasible-interior-point methods for degenerate LCP*, edited by J. Renegar, M. Shub and S. Smale. AMS, Providence, RI. *Math. Numer. Anal., Lect. Appl. Math.* **32** (1996) 443-454.
- [8] J. Ji, A. Potra and S.Huang, Predictor-corrector method for linear complementarity problems with polynomial complexity and superlinear convergence. *JOTA* **85** (1995) 187-199.
- [9] A. Kadiri, *Analyse numérique des méthodes de points intérieurs pour les problèmes de complémentarité linéaire et la programmation quadratique convexe*. Thèse de Doctorat, INSA de Rouen (2001).
- [10] C.T. Kelley, Iterative methods for linear and nonlinear equations. *Frontiers Appl. Math.* **16** (1995).
- [11] M. Kojima, S. Mizuno and A. Yoshise, A polynomial-time algorithm for a class of linear complementarity problems. *Math. Program.* **44** (1989) 1-26.
- [12] M. Kojima, Y. Kurita and S. Mizuno, Large-step interior point algorithms for linear complementarity problems. *SIAM J. Optim.* **3** (1993) 398-412.
- [13] K. Kortanek and J. Zhu, New purification algorithms for linear programming. *Naval Res. Logist* **35** (1988) 571-583.
- [14] K. Mcshane, Superlinearly convergent $O(\sqrt{n}L)$ -iteration interior-point algorithms for LP and the monotone LCP. *SIAM J. Optim.* **4** (1994) 247-261.
- [15] R. Monteiro and I. Adler, Interior path-following primal-dual algorithms, part II: Convex quadratic programming. *Math. Program.* **44** (1989) 43-66.
- [16] R. Monteiro and S. Wright, Local convergence of interior-point algorithms for degenerate monotone LCP. *Comput. Optim. Appl.* **3** (1994) 131-155.

- [17] C.R. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, New Jersey (1982).
- [18] F.A. Potra and R. Sheng, A superlinearly convergent infeasible-interior-point algorithm for degenerate LCP. *J. Optim. Theory Appl.* **97** (1998) 249-269.
- [19] Y. Ye, On the finite convergence of interior point algorithms for linear programming. *Math. Program.* **57** (1992) 325-335.
- [20] Y. Ye, *Interior Point Algorithms: Theory and Analysis*. John Wiley, New York (1997).
- [21] Y. Ye and K.M. Anstreicher, On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for LCP. *Math. Program.* **62** (1993) 537-551.