Résolution d'un problème de programmation quadratique à variables bornées avec une M-matrice

Katia HASSAINI et Mohand Ouamer BIBI

Unité de Recherche LaMOS
Université de Béjaia 06000, Algérie,
Hassaini@Gmail.com mobibi.dz@Gmail.com

Résumé Dans cette article, une méthode de résolution d'un problème de programmation quadratique à variables bornées avec une M-matrice est présentée. Elle se base sur les algorithmes de Luk et Pagano [2], Stachurski [4], ainsi que sur l'algorithme de Voglis et Lagaris [7]. Les deux premiers utilisent le fait qu'une M-matrice possède une inverse non négative qui permet d'avoir une suite monotone de solutions réalisables, tandis que le troisième s'inspire d'une méthode de points extérieurs [6]. En introduisant le concept de support pour une fonction objectif [17], notre approche se différencie par une condition plus générale qui permet d'avoir une pseudo-solution liée à un support coordinateur. La programmation sous MATLAB de notre méthode nous a permis de faire une comparaison numérique avec la méthode d'activation des contraintes (ASM) existante dans la Toolbox SVM de Matlab, et de les illustrer par deux exemples.

Mots clés Programmation quadratique convexe, M-matrices, méthode de Newton projetée, méthode de support, comparaison numérique.

1 Introduction

Dans la littérature, plusieurs approches ont été proposées pour la résolution des problèmes de programmation quadratique quand la matrice D associée est définie positive. Cependant, il est possible d'exploiter les propriétés spéciales d'une M-matrice pour obtenir des algorithmes spéciaux plus efficaces. D'ailleurs, de tels problèmes avec des M-matrices trouvent des applications dans la résolution numérique des équations aux dérivées partielles elliptiques. Ces problèmes incluent divers types de problèmes de Dirichlet avec obstacles [11,12], les modèles d'application de torsion à une barre [20], etc. Les M-matrices sont aussi connues pour avoir de nombreuses applications dans la modélisation des systèmes dynamiques, dans les sciences économiques et écologiques [13,14,15]. Plusieurs de leurs propriétés sont utilisées pour établir des résultats de stabilité pour les systèmes dynamiques en général [8,9].

L'objet de ce travail est le développement d'une nouvelle méthode pour la résolution des problèmes de programmation quadratique à variables bornées avec une M-matrice. En s'inspirant des travaux de Luk et Pagano [2], Stachurski [4], ainsi que sur l'algorithme de Voglis et Lagaris [7], et en introduisant le concept de support pour une fonction objectif [17], nous avons pu construire une méthode qui se différencie des travaux précités par une condition plus générale qui permet d'avoir une pseudo-solution liée à un support coordinateur.

2 Position du problème et définitions

Considérons le problème de programmation quadratique à variables bornées suivant :

$$\begin{cases} F(x) = \frac{1}{2}x^T Dx + c^T x \longrightarrow \min, \\ \qquad\qquad\qquad\qquad l \leq x \leq u, \end{cases} \tag{1}$$

où $c = c(J) = (c_j, \quad j \in J)$, $l = l(J) = (l_j, \quad j \in J)$, $u = u(J) = (u_j, \quad j \in J)$ et $x = x(J) = (x_j, \quad j \in J)$ sont des n-vecteurs réels, avec $J = \{1, 2, \cdots, n\}$. La matrice $D = D(J, J)$ est une M-matrice carrée symétrique $\quad (D = D^T)\ $ d'ordre n. Le symbole $(^T)$ est celui de la transposition.

Rappelons qu'une M-matrice $D = (d_{ij},\ 1 \leq\ i,\ j \leq n)$ satisfait les conditions suivantes [2,15] :

$$d_{ii} > 0, \qquad\qquad d_{ij} \leq 0, \quad i \neq j, \qquad\qquad D^{-1} \geq 0,$$

où le symbole $D^{-1} \geq 0$ veut dire que tous les éléments de la matrice inverse D^{-1} sont positifs ou nuls. De plus, Une M-matrice symétrique est toujours définie positive $(x^T Dx > 0,\ \forall\ x \neq 0)$ et toute sous-matrice carrée symétrique d'une M-matrice est une M-matrice.

Définition 1.
Un vecteur x tel que $l \leq x \leq u$ est appelé *solution réalisable* du problème (1). Une solution réalisable x^0 est dite *optimale* si elle réalise le minimum de la fonction objectif du problème (1).

Ainsi, nous avons

Théorème 1. *Une solution réalisable x^0 du problème (1) est optimale si et seulement si les conditions suivantes sont satisfaites :*

$$\begin{cases} x_j^0 = l_j & \Rightarrow \quad g_j(x^0) \geq 0, \\ x_j^0 = u_j & \Rightarrow \quad g_j(x^0) \leq 0, \\ l_j < x_j^0 < u_j & \Rightarrow \quad g_j(x^0) = 0, \quad j \in J, \end{cases} \tag{2}$$

où $g(x) = g(J) = Dx + c$ est le gradient de la fonction objectif F au point x.

Considérons le problème de programmation quadratique sans contraintes :

$$\begin{cases} F(x) = \frac{1}{2}x^T Dx + c^T x \longrightarrow \min, \\ \qquad\quad x \in \mathbb{R}^n. \end{cases} \tag{3}$$

La solution optimale $\widehat{x}$ du problème (3) vérifie

$$D\widehat{x} + c = 0 \iff \widehat{x} = -D^{-1}c.$$

Soient J_S et J_N une partition de J : $J_S \cup J_N = J, \quad J_S \cap J_N = \emptyset$. Le gradient de la fonction objectif F au point x peut alors s'écrire sous la forme suivante :

$$g = \begin{pmatrix} g_S \\ g_N \end{pmatrix}, \quad g_S = g(J_S) = D_S x_S + D_{SN} x_N + c_S, \quad g_N = g(J_N) = D_{NS} x_S + D_N x_N + c_N,$$

où

$$x = \begin{pmatrix} x_S \\ x_N \end{pmatrix},\ c = \begin{pmatrix} c_S \\ c_N \end{pmatrix},\ D_S = D(J_S, J_S),\ D_N = D(J_N, J_N),\ D_{SN} = D(J_S, J_N).$$

Pour tout sous-ensemble J_S dans J, la condition suivante est vérifiée :

$$\det D_S = \det D(J_S, J_S) \neq 0.$$

Définition 2.
Le sous-ensemble J_S est appelé *support* de la fonction objectif F et le couple $\{x, J_S\}$, formé d'une solution réalisable x et du support J_S, est appelé *solution réalisable de support.*

Définition 3.
- Soient J_+ et J_- des ensembles d'indices tels que : $J_+ \cup J_- = J_N, \quad J_+ \cap J_- = \emptyset$. Un vecteur $\kappa = \kappa(J) = \Big(\kappa(J_S), \kappa(J_+), \kappa(J_-)\Big)$ vérifiant

$$\begin{cases} \kappa(J_+) = l(J_+), \\ \kappa(J_-) = u(J_-), \\ \kappa_S = -D_S^{-1}[c_S + D_{SN}\kappa_N]\ , \end{cases}$$

est dit *pseudo-solution* du problème(1). Une pseudo-solution κ vérifie $g_S(\kappa) = 0$. De plus, si κ vérifie l'inégalité $l(J_S) \leq \kappa(J_S) \leq u(J_S)$, alors κ sera une solution réalisable.
- Le support $J_p = \{J_S, J_+, J_-\}$ est appelé *support coordinateur* s'il existe une pseudo-solution κ telle que :

$$g_j(\kappa) \geq 0, \qquad j \in J_+, \tag{4}$$
$$g_j(\kappa) \leq 0, \qquad j \in J_-. \tag{5}$$

On dit dans ce cas que la pseudo-solution κ est associée au support coordinateur J_P.

Théorème 2. *Une pseudo-solution κ associée à un support coordinateur J_p est optimale dans le problème (1) si et seulement si*

$$l_j \leq \kappa_j \leq u_j, \qquad j \in J_S. \tag{6}$$

Preuve 1. *Le vecteur κ est une solution réalisable et satisfait les conditions d'optimalité (2).*$\blacksquare$

Remarque 1. Comme $g(\widehat{x}) = 0$, alors la solution optimale $\widehat{x}$ du problème sans contraintes (3) est une pseudo-solution du problème (1), associée à un support coordinateur $J_p = \{J_S, J_+, J_-\}$, avec $J_S = J$ et $J_+ = J_- = \emptyset$. D'aprés le théorème 2, si $l \leq \widehat{x} \leq u$, alors $x^0 = \widehat{x}$ est une solution optimale du problème (1).

Considérons le dual du problème primal (1) :

$$\varphi(\kappa) = -\frac{1}{2}\kappa^T D\kappa + v^T l - w^T u \longrightarrow \max, \tag{7}$$

$$D\kappa + c - v + w = 0, \tag{8}$$

$$\lambda = (\kappa, v, w), \kappa \in \mathbb{R}^n,\ v \geq 0,\ w \geq 0. \tag{9}$$

Définition 4.

- Tout triplet $\lambda = (\kappa, v, w) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ vérifiant (8) et (9) est appelé *solution réalisable duale* du problème (7)-(9).
- Soit $J_P = \{J_S, J_+, J_-\}$ un support coordinateur du problème (1). Alors le vecteur $\lambda = (\kappa, v, w)$ défini par

$$\begin{cases} v_j = g_j(\kappa), & w_j = 0 & j \in J_+, \\ v_j = 0, & w_j = -g_j(\kappa) & j \in J_-, \\ v_j = w_j = 0, & & j \in J_S, \end{cases}$$

est dit *solution réalisable accordée* du problème dual (7)-(9).

Lemme 1. *[16] Pour toute solution réalisable duale quelconque $\lambda = (\kappa, v, w)$, il existe une solution réalisable duale accordée $\overline{\lambda} = (\overline{\kappa}, \overline{v}, \overline{w})$ telle que*

$$\varphi(\lambda) \leq \varphi(\overline{\lambda}). \tag{10}$$

En vertu de ce lemme, on ne s'interesse dorénavant qu'aux solutions réalisables duales accordées.

3 Algorithme de la méthode

Poser $k = 0$ et calculer la pseudo-solution $\kappa^{(0)}$ du problème(1) :

$$g^{(0)}(\kappa^{(0)}) = D\kappa^{(0)} + c = 0 \Longrightarrow \kappa^{(0)} = -D^{-1}c.$$

(1) Si $\kappa^{(0)}$ est tel que $l \leq \kappa^{(0)} \leq u$, alors terminer et le vecteur $x^0 = \kappa^{(0)}$ est la solution optimale du problème (1).

(2) Sinon, définir les ensembles $J_S^{(k)}$, $J_+^{(k)}$, $J_-^{(k)}$ et $J_N^{(k)}$ tels que :

$$J_+^{(k)} = \{j \in J : \ \ \kappa_j^{(k)} < l_j, \ ou \ \ \ \kappa_j^{(k)} = l_j, \ \ g_j^{(k)} \geq 0\},$$
$$J_-^{(k)} = \{j \in J : \ \ \kappa_j^{(k)} > u_j, \ ou \ \ \ \kappa_j^{(k)} = u_j, \ \ g_j^{(k)} \leq 0\}, \ \ J_N^{(k)} \ = \ (J_+^{(k)} \cup J_-^{(k)}),$$
$$J_S^{(k)} = \{j \in J : \ \ l_j < \kappa_j^{(k)} < u_j, \ ou \ \ \kappa_j^{(k)} = l_j \ \ et \ \ g_j^{(k)} < 0, \ ou \ \ \kappa_j^{(k)} = u_j \ \ et \ \ g_j^{(k)} > 0\},$$
$$J = J_N^{(k)} \ \cup J_S^{(k)}.$$

(3) Construire $\kappa^{(k+1)}$ de la manière suivante :

$$\begin{cases} \kappa_j^{(k+1)} = l_j, \quad\ j \in J_+^{(k)}, \\ \kappa_j^{(k+1)} = u_j, \quad\ j \in J_-^{(k)}, \\ \kappa^{(k+1)}(J_S^{(k)}) = -D^{-1}(J_S^{(k)}, J_S^{(k)})\big[c(J_S^{(k)}) + D\big(J_S^{(k)}, J_N^{(k)}\big)\kappa^{(k+1)}(J_N^{(k)})\big] \ . \end{cases}$$

(4) Calculer

$$g_N^{(k+1)}(\kappa^{(k+1)}) = D(J_N^{(k)}, J_S^{(k)})\kappa^{(k+1)}(J_S^{(k)}) + D(J_N^{(k)}, J_N^{(k)})\kappa^{(k+1)}(J_N^{(k)}) + c(J_N^{(k)}).$$

(5) Vérifier si la nouvelle pseudo-solution est optimale

(i) *Si* $l_j \leq \kappa_j^{(k+1)} \leq u_j, \ \forall j \in J_S^{(k)}$ *et* $g_j^{(k+1)} \geq 0, \ \forall \ j \in J_+^{(k)}$ *et* $g_j^{(k+1)} \leq 0, \ \forall \ j \in J_-^{(k)}$, alors stop, la solution est $x^0 = \kappa^{(k+1)}$.

(ii) Sinon, poser $k = k + 1$ et aller vers (2).

4 Exemple Numérique

Considérons le problème de programmation quadratique à variables bornées suivant :

$$\begin{cases} F(x) = \sum_{i=1}^{5} x_i^2 - x_1x_2 - x_2x_3 - x_3x_4 - x_4x_5 - x_1 + 2x_2 + 7x_3 - 3x_4 + 4x_5 \longrightarrow \text{min}, \\ \\ -4 \leq x_1 \leq 4, \ 0 \leq x_2 \leq 5, \ -2 \leq x_3 \leq 3, \ 1 \leq x_4 \leq 6, \ -3 \leq x_5 \leq 3, \end{cases} \tag{11}$$

Ce problème s'écrit sous la forme (1), avec les données suivantes :

$$D = D(J, J) = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}, \qquad c = c(J) = \begin{pmatrix} -1 \\ 2 \\ 7 \\ -3 \\ 4 \end{pmatrix}$$

$$l = l(J) = \begin{pmatrix} -4 \\ 0 \\ -2 \\ 1 \\ -3 \end{pmatrix}, \quad u = u(J) = \begin{pmatrix} 4 \\ 5 \\ 3 \\ 6 \\ 3 \end{pmatrix}, \quad J = \{1, 2, 3, 4, 5\}.$$

– On pose $k = 0$, et on calcule la pseudo-solution κ^0 du prolème (1) :

$$g^{(0)}(\kappa^{(0)}) = D\kappa^{(0)} + c = 0 \Rightarrow \kappa^{(0)} = D^{-1}c,$$
$$\Rightarrow \kappa^{(0)} = \begin{pmatrix} 5/6 & 2/3 & 1/2 & 1/3 & 1/6 \\ 2/3 & 4/3 & 1 & 2/3 & 1/3 \\ 1/2 & 1 & 3/2 & 1 & 1/2 \\ 1/3 & 2/3 & 1 & 4/3 & 2/3 \\ 1/6 & 1/3 & 1/2 & 2/3 & 5/6 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 7 \\ -3 \\ 4 \end{pmatrix},$$
$$\Rightarrow \kappa^{(0)} = \begin{pmatrix} -11/3 \\ -25/3 \\ -11 \\ -20/3 \\ -16/3 \end{pmatrix}.$$

$\kappa^{(0)}$ n'est pas réalisable alors :

<u>Itération 1 :</u>

– On définit les ensembles $J_S^{(0)},\ J_+^{(0)},\ J_-^{(0)}\ et\ \ J_N^{(0)}$ tels que :

$$J_+^{(0)} = \{j \in J : \kappa_j^{(0)} < l_j,\ ou\ \ \kappa_j^{(0)} = l_j,\ \ g_j^{(0)} \geq 0\} = \{2,3,4,5\},$$
$$J_-^{(0)} = \{j \in J : \kappa_j^{(0)} > u_j,\ ou\ \ \kappa_j^{(0)} = u_j,\ \ g_j^{(0)} \leq 0\} = \emptyset,$$
$$J_S^{(0)} = \{j \in J : u_j < \kappa_j^{(0)} < l_j\ ou\ \ \kappa_j^{(0)} = l_j\ \ et\ \ g_j^{(0)} < 0,\ ou\ \ \kappa_j^{(0)} = u_j\ \ et\ \ g_j^{(0)} > 0\} = \{1\},$$
$$J_N^{(0)} = J_+^{(0)} \cup J_-^{(0)} = \{2,3,4,5\},$$

– On construit $\kappa^{(1)}$ de la manière suivante :

$$\begin{cases} \kappa_j^{(1)} = l_j, & j \in J_+^{(0)}, \\ \kappa_j^{(1)} = u_j, & j \in J_-^{(0)}, \\ \kappa^{(1)}(J_S^{(0)}) = -D^{-1}(J_S^{(0)}, J_S^{(0)})\big[c(J_S^{(0)}) + D\big(J_S^{(0)}, J_N^{(0)}\big)\kappa^{(1)}(J_N^{(0)})\big]. \end{cases} \tag{12}$$

$$\kappa_j^{(1)} = l_j = \begin{pmatrix} \kappa_2^{(1)} \\ \kappa_3^{(1)} \\ \kappa_4^{(1)} \\ \kappa_5^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 1 \\ -3 \end{pmatrix}, \qquad j \in J_+^{(0)},\ \ et$$

$$\kappa^{(1)}(J_S^{(0)}) = \kappa_1^{(1)} = -\frac{1}{2}\Big[-1 + (-1,0,0,0)\begin{pmatrix} 0 \\ -2 \\ 1 \\ -3 \end{pmatrix}\Big] = 1/2.$$

On a donc $\kappa^{(1)} = \begin{pmatrix} 1/2 \\ 0 \\ -2 \\ 1 \\ -3 \end{pmatrix}$

– On calcule

$$g_N^{(1)}(\kappa^{(1)}) = D(J_N^{(0)}, J_S^{(0)})\kappa^{(1)}(J_S^{(0)}) + D(J_N^{(0)}, J_N^{(0)})\kappa^{(1)}(J_N^{(0)}) + c(J_N^{(0)}).$$

$$g_N^{(1)}(\kappa^{(1)}) = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \frac{1}{2} + \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ -2 \\ 1 \\ -3 \end{pmatrix} + \begin{pmatrix} 2 \\ 7 \\ -3 \\ 4 \end{pmatrix} = \begin{pmatrix} 7/2 \\ 2 \\ 4 \\ -3 \end{pmatrix}$$

– Si $l_j \leq \kappa_j^{(1)} \leq u_j, \ \forall \ j \in J_S^{(0)}$ et $g_j^{(1)} \geq 0, \ \forall \ j \in J_+^{(0)}$ et $g_j^{(1)} \leq 0, \ \forall \ j \in J_-^{(0)}$,non, alors la solution $\kappa^{(1)}$ n'est pas optimale, on pose $k = k + 1 = 1$.

<u>Itération 2 :</u>

– On redéfinit les ensembles $J_S^{(1)}, \ J_+^{(1)}, \ J_-^{(1)} \ \ et \ J_N^{(1)}$ tels que :

$$\begin{aligned}
J_+^{(1)} &= \{j \in J : \kappa_j^{(1)} < l_j, \ ou \ \ \kappa_j^{(1)} = l_j, \ \ g_j^{(1)} \geq 0\} = \{2, 3, 4\}, \\
J_-^{(1)} &= \{j \in J : \kappa_j^{(1)} > u_j, \ ou \ \ \kappa_j^{(1)} = u_j, \ \ g_j^{(1)} \leq 0\} = \emptyset, \\
J_S^{(1)} &= \{j \in J : u_j \leq \kappa_j^{(1)} \leq l_j\} = \{1, 5\}, \\
J_N^{(1)} &= J_+^{(1)} \cup J_-^{(1)} = \{2, 3, 4\},
\end{aligned}$$

– On construit $\kappa^{(2)}$ selon (12) :

$$\kappa^{(2)}(J_+^{(1)}) = \begin{pmatrix} \kappa_2^{(2)} \\ \kappa_3^{(2)} \\ \kappa_4^{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix} et$$

$$\kappa^{(2)}(J_S^{(1)}) = \begin{pmatrix} \kappa_1^{(2)} \\ \kappa_5^{(2)} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \left[\begin{pmatrix} -1 \\ 4 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix} \right],$$

$$\kappa^{(2)}(J_S^{(1)}) = \begin{pmatrix} \kappa_1^{(2)} \\ \kappa_5^{(2)} \end{pmatrix} = \begin{pmatrix} 1/2 \\ -3/2 \end{pmatrix}$$

D'où

$$\kappa^{(2)} = \begin{pmatrix} 1/2 \\ 0 \\ -2 \\ 1 \\ -3/2 \end{pmatrix}$$

– On calcule $g^{(2)}(J_N^{(1)})$ du $\kappa^{(2)}(J_N^{(1)})$

$$g_N^{(2)}(\kappa^{(2)}) = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1/2 \\ -3/2 \end{pmatrix} + \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 7 \\ -3 \end{pmatrix} = \begin{pmatrix} 7/2 \\ 2 \\ 5/2 \end{pmatrix}$$

– On vérifie si $\kappa^{(2)}$ est optimale, Si $l_j \leq \kappa_j^{(2)} \leq u_j, \ \forall\, j \in J_S^{(1)}$ et $g_j^{(2)} \geq 0, \ \forall\, j \in J_+^{(1)}$ et $g_j^{(2)} \leq 0, \ \forall\, j \in J_-^{(1)}$, oui , alors terminer, la solution optimale du problème (11) est donc $x^0 = \kappa^{(2)} = (1/2, 0, -2, 1, -3/2)^T$, avec $F(x^0) = -25/2$.

5 Comparaisons numériques sous Matlab

Nous avons choisi deux problèmes représentatifs. Le but est d'examiner l'efficacité de notre approche nommée Box-QP et de faire une comparaison numérique avec la méthode d'activation des contraintes (ASM) existante dans la Toolbox SVM de Matlab. Notre approche a été implémentée sous l'environnement Matlab version 7.11 et nous avons utilisé un PC portable ayant une RAM de 4GO, une CPU de 2,30 Ghz, l'ordinateur étant doté du système d'exploitation Windows7. Les critères de comparaison entre les deux méthodes sont le temps (CPU) en seconde et le nombre d'itérations (Iter) nécessaire pour obtenir l'optimum.

5.1 Example 1.

Considérons le programme quadratique (1) :

$$\begin{cases} F(x) = \frac{1}{2} x^T D x + c^T x \longrightarrow \min, \\ \qquad\qquad\qquad l \leq \ x \leq u. \end{cases}$$

La matrice D choisie est la matrice correspondante à la discrétisation par différences finies du problème de Dirichlet en dimension 1 :

$$D \ = \begin{pmatrix} 2 & -1 & & & \bigcirc \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ \bigcirc & & & -1 & 2 \end{pmatrix}_{n \times n} . \qquad (13)$$

Les vecteur $c, \ l$ et u sont générés de la manière suivante :

$$\begin{cases} c_j \ = 11 - 23 \ r_j, \\ l_j \ = 8 - 20 \ r_j, \\ u_j = 11 - 20 \ r_j, \quad j = \overline{1, n}. \end{cases} \qquad (14)$$

Le nombre r_j est un nombre aléatoire qui suit une loi uniforme $r_j \ \in \ U\ [0, 1]$. Les temps machine d'exécution (CPU) et le nombre d'itérations (Iter) de chaque programme pour la résolution du programme quadratique sont représentés dans le tableau ci-dessous :

Dimension	Box-QP		ASM	
n	Iter	CPU	Iter	CPU
200	05	0.0073	13	0.2606
400	06	0.0363	13	0.2523
600	06	0.0768	14	0.2915
1000	07	0.0719	13	0.3097
1200	07	0.0854	14	0.3319
1400	08	0.1108	15	0.3818
1600	08	0.1085	16	0.4104
1800	06	0.1380	16	0.4481
2000	07	0.1569	16	0.4879
2500	07	0.2122	11	0.6040
3000	07	0.2731	15	0.8373
3500	07	0.3409	13	1.0311
4000	08	0.4553	16	1.2727
4500	07	0.5001	16	1.5787
5000	08	0.6364	16	1.8631

Table 1. Résultats de comparaison entre Box-QP et ASM

5.2 Example 2.

Dans cet exemple, on choisit la matrice D comme étant une matrice déduite de l'approximation du Laplacien par des différences finies en 5-points :

$$D \;=\; \begin{pmatrix} B & -I & & & \bigcirc \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ \bigcirc & & & -I & B \end{pmatrix}_{m^2\times m^2} , \tag{15}$$

où

$$B \;=\; \begin{pmatrix} 4 & -1 & & & \bigcirc \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ \bigcirc & & & -1 & 4 \end{pmatrix}_{m\times m} . \tag{16}$$

Avec $n = m^2$. Les vecteurs c, l et u sont générés selon (14). Les résultats obtenus sont représentés dans le tableau ci-dessous :

6 Conclusion

Dans cette article, nous avons proposé une nouvelle approche pour la résolution d'un problème de programmation quadratique à variables bornées avec une M-matrice, et ce, en introduisant le concept de support coordinateur pour le problème (1). La programmation sous MATLAB de notre méthode nous a permis de faire

Dimension	Box-QP		ASM	
n	Iter	CPU	Iter	CPU
200	06	0.0091	13	0.2724
400	03	0.0219	13	0.2243
600	03	0.0281	15	0.2467
800	03	0.0358	13	0.2627
1000	03	0.0438	14	0.2970
1400	04	0.0637	14	0.3602
1600	03	0.0726	16	0.4219
1800	03	0.0845	15	0.4383
2000	03	0.0994	13	0.4894
2500	04	0.1504	15	0.6407
3000	03	0.1748	16	0.8372
3500	03	0.2246	14	1.0090
4000	03	0.2792	15	1.2440
4500	03	0.3479	17	1.6073
5000	03	0.4253	14	1.8248
5200	03	0.4399	15	1.9860

Table 2. Résultats de comparaison entre Box-QP et ASM

une comparaison numérique avec la méthode d'activation des contraintes (ASM) existante dans la Toolbox SVM de Matlab, et de les illustrer par deux exemples.

Références

1. Chandrasekaran, R. : A special case of the complementary pivot problem. Opsearch, **7** (1970), 263–268.
2. Luk, F. T. and Pagano, M. : Quadratic programming with M-Matrices. Linear Algebra And Its Applications, **33** (1980), 15–40.
3. Stachurski, A. : Monotone sequences of feasible solutions for quadratic programming problems with M-matrices and box constraints. In System Modeling and Optimization. Book series : Lecture Notes in Control and Information Sciences, (1986), Vol 84, 896–902.
4. Stachurski, A. : An Equivalence between two algorithms for a class of quadratic programming problems with M-matrices. Optimization, **21(6)** (1990), 871–878.
5. Li, L. and Kobayashi, Y. : A block recursive algorithm for the linear complementarity problem with an M-matrix. International Journal of Innovative, Computing, Information and Control, Vol 2, Number 6, (2006), 1327–1335.
6. Kunisch, K. and Rendl, F. : An infeasible active set method for convex problems with simple bounds. SIAM Journal on optimization, **14(1)**(2003), 35–52.
7. Voglis, C. and Lagaris, I. E. : BOXCQP : An Algorithm for Bound Constrained Convex Quadratic Problems. 1st International Conference "From Scientific Computing To Computational Engineering", Vol 1, (2004), 261–268, Athens, Greece, September 8-10.
8. Stipanovic, D. M. and Šiljak, D. D. : Stability of polytopic systems via convex M-matrices and parameter-dependent Liapunov functions. Nonlinear Analysis, **40** (2000), 589–609.

9. Stipanovic, D. M., Shankaran, S. and Tomlin, C.J. : Multi-agent avoidance control using an M-matrix property. Electronic Journal of Linear Algebra. A publication of the International Linear Algebra Society, **12** (2005), 64–72.
10. Pang, J. S. : On a class of least-element complementarity problems.(1976), Report SOL 76-10, Systems Optimization Lab, Stanford Univ.
11. Levati, G., Scarpini, F. and Volpi, G. : Sul trattemento numerico di alcuni problemi variazionali di tipo unilaterale. L.A.N. Pub.82, Univ. of Pavia (1974).
12. Scarpini, F. : Some algorithms solving the unilateral Dirichlet problems with two constraints. Calcolo, **12**(1975), 113–149.
13. Šiljak, D. D. : Large-scale Dynamic Systems : Stability and Structure. North-Holland, New York, (1978).
14. Varga, R. S. : Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, N.J. (1962).
15. Berman, A. and Plemmons, R. J. : Nonnegative Matrices in Mathematical Sciences. Academic Press, New York, (1979).
16. Gabasov, R., Kirillova, F. M., Kostyukova, O. I. and Raketsky, V.M. : Constructive methods of optimization, volume 4 : Convex Problems. University Press, Minsk (1987).
17. Brahmi, B. and Bibi, M. O. : Dual Support method for solving convex quadratic programs. OPTIMIZATION, 2010, Vol 59, n° 6, p. 851–872.
18. Bibi, M. O. and Radjef, S. : Solution of a convex quadratic program with mixed variables via the direct support method. In Actes du Colloque MOAD : Méthodes et Outils d'Aide à la Décision, 18-20 Novembre 2007, Université de Béjaia, p.39-44.
19. Hassaini, K. and Bibi, M. O. :Méthode adaptée pour la résolution d'un problème de programmation quadratique avec des M-matrices. In Actes du Colloque sur l'Optimisation et les Systèmes d'Information, 18-20 Avril 2010, Université de Ouargla, pp.123–134. (in French).
20. Céa, J. and Glowinski, R. : Sur des méthodes d'optimisation par relaxation. RAIRO, **R-3** (1973) 5–32. (in French).
21. Bertsekas, D. P. : Projection Newton Method for Optimization Problems with Simple Constraints. SIAM Journal on Control and Optimization, **20**(1982)2, 221–246.