

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ FERHAT ABBAS SÉTIF-1

FACULTÉ DES SCIENCES

DÉPARTEMENT DE MATHÉMATIQUES



THÈSE

Présentée par :

BOUDJELLAL NAWEL

Pour obtenir le titre de Doctorat LMD 3ème cycle

OPTION : OPTIMISATION ET CONTRÔLE

THÈME

**MÉTHODES DE POINTS INTÉRIEURS POUR LA PROGRAMMATION
QUADRATIQUE CONVEXE :
THÉORIE, ALGORITHMES ET APPLICATIONS**

Soutenue le : 24/12/2020

Devant le jury composé de :

M. Bachir MERIKHI	U.F.A. Sétif-1	Prof.	Président
Mme. Hayet ROUMILI	U.F.A. Sétif-1	Prof.	Encadreur de thèse
M. Djamel BENTERKI	U.F.A. Sétif-1	Prof.	Co-encadreur de thèse
M. Abdessamed AMIR	U.M.A.B. Mostaganem	Prof.	Examineur
M. Mousaab BOUAFIA	U. 8 Mai 1945 Guelma	M.C.A	Examineur

2020/2021

Dédicaces

Je dédie ce modeste travail :

À mes chers parents qui m'ont toujours encouragés et soutenus dans la poursuite de mes études et durant toutes les années de recherche, un grand merci à vous.

À mes adorables sœurs : Samia, Khadidja et Nour El-Hoda.

À mon frère Younes.

À mon neveu Mouataz.

À ma famille et toutes mes chères amies.

Nawel

Remerciements

En tout premier lieu, je remercie **Dieu** le tout puissant, de m'avoir donné assez de courage pour dépasser toutes les difficultés et accomplir ce travail.

Je tiens à exprimer ma profonde reconnaissance envers mon directeur de thèse et co-directeur de thèse, Madame **Hayet Roumili** Professeur à l'Université Ferhat Abbas-Sétif-1 et Monsieur **Djamel Benterki** Professeur à l'Université Ferhat Abbas-Sétif-1, pour leur confiance, leurs conseils, leur aide inestimable et leur patience conjugée durant les années de recherche.

Je remercie tout particulièrement Monsieur **Bachir Merikhi** Professeur à l'Université Ferhat Abbas-Sétif-1, pour l'honneur qu'il m'a fait en acceptant de présider le jury de cette thèse.

Je suis très honoré par la présence dans le jury des Messieurs : **Abdessamad Amir** Professeur à l'Université Abdelhamid Ibn Badis, Mostaganem et **Moussaab Bouafia** Maître de Conférences A à l'Université 8 mai 1945 Guelma qui ont bien voulu être juges ce travail. Je leur adresse toute ma reconnaissance.

Je tiens à témoigner toute ma reconnaissance aux Messieurs suivants, pour leur aide, leurs conseils et leurs paroles de motivation à la réalisation de cette thèse :

Monsieur **Mohamed Achache**, Professeur à l'Université Ferhat Abbas-Sétif 1.

Monsieur **Nadhir Chougui**, mon directeur de mémoire de master 2, Maître de Conférences B à l'Université Ferhat Abbas-Sétif-1.

Mes remerciements s'adressent également à toute l'équipe administrative de département de mathématiques de l'Université Ferhat Abbas-Sétif-1.

Enfin que tous ceux qui n'ont pas été mentionnés et qui ont contribué à la réalisation, de près ou de loin, de ce travail reçoivent mes remerciements.

TABLE DES MATIÈRES

Introduction générale	v
Notations principales	x
1 Notions d'analyse convexe et optimisation quadratique	1
1.1 Notions d'analyse convexe	1
1.1.1 Ensembles et fonctions convexes	2
1.1.2 Convexité et différentiabilité	3
1.1.3 Notations asymptotiques et vitesse de convergence	3
1.2 Programmation mathématique	4
1.2.1 Position du problème et sa classification	4
1.2.2 Principaux résultats d'existence et d'unicité d'un optimum . . .	5
1.2.3 Qualification des contraintes	5
1.2.4 Conditions d'optimalité	6
1.2.5 Dualité au sens de Lagrange	7
1.3 Programmation quadratique convexe	8
1.3.1 Problème quadratique convexe primal	9
1.3.2 Problème quadratique convexe dual	9
1.3.3 Dualité en programmation quadratique	10
1.3.4 Méthodes de résolution d'un (PQC)	11

2	Survol sur les fonctions noyaux adaptées aux méthodes de points intérieurs primales-duales de type trajectoire centrale	29
2.1	Fonctions noyaux et leurs propriétés	30
2.1.1	Conditions de qualification d'une fonction noyau	31
2.1.2	Propriétés des fonctions noyaux	35
2.2	Présentation de la méthode	37
2.3	Description algorithmique	41
2.4	Analyse de l'algorithme	42
2.4.1	Borne supérieure de la fonction barrière $\Phi(v)$ pendant une itération externe	43
2.4.2	Détermination du pas de déplacement α et la décroissance de $\Phi(v)$ pendant une itération interne	45
2.5	Complexité algorithmique	52
3	Méthode de points intérieurs de type trajectoire centrale basée sur des nouvelles fonctions noyaux pour (PQC)	55
3.1	Nouvelle fonction noyau avec terme barrière exponentiel	57
3.1.1	Propriétés de la nouvelle fonction noyau	60
3.1.2	Analyse de l'algorithme	64
3.1.3	Tests numériques	68
3.2	Nouvelle fonction noyau avec terme barrière polynomial (de Laurent)	77
3.2.1	Eligibilité de la nouvelle fonction noyau	77
3.2.2	Complexité de l'algorithme	83
3.2.3	Tests numériques	84
	Conclulsion générale	94
	Bibliographie	97

INTRODUCTION GÉNÉRALE

La programmation mathématique se propose pour objet l'étude théorique des problèmes d'optimisation ainsi que la conception et la mise en œuvre des algorithmes de résolution. Ses applications sont extrêmement nombreuses et variées, que ce soit dans la physique (minimisation d'énergie), l'industrie (optimisation de la qualité de production), l'économie (problèmes de localisation, de gestion de production et de distribution, de logistique et de transport, de gestion de stocks, de tarification, d'optimisation de flux dans les réseaux), la finance (optimisation de portefeuille), le traitement d'image, la biologie, les sciences de l'ingénieur, la géométrie (la petite boule englobante). . . etc.

Les méthodes de points intérieurs (*MPIs*) sont des méthodes effectives pour résoudre pas mal de problèmes d'optimisation. Ces méthodes reposent sur la fonction barrière logarithmique définie, en 1955, par Frisch [34]. Cette fonction barrière logarithmique est largement étudiée dans les années soixante, notamment dans le livre de Fiacco et McCormick [31]. C'est dans ce livre que le terme de points intérieurs est introduit. Les itérés générés par les méthodes basées sur cette fonction sont strictement réalisables : ils restent à l'intérieur du domaine réalisable, d'où la terminologie de ces méthodes.

Avant 1984, tout problème d'optimisation linéaire se résolvait par la méthode du simplexe développée par Dantzig [24] ou par une variante de celle-ci. Des recherches ont été menées pour mettre au point une autre méthode mais aucune de celles proposées n'améliorait celle du simplexe.

Dans les années 70, la théorie de la complexité devient une part intégrante de

l'optimisation linéaire. Lorsque Klee et Minty [47] ont trouvé un exemple qui montrait pour la première fois que la méthode du simplexe peut prendre un nombre exponentiel d'itérations. Une question demeure toutefois ouverte : existe-t-il un algorithme de complexité polynomiale pour résoudre un programme linéaire ?, c'est à dire de résoudre le problème en un nombre d'opérations qui doit être borné par un polynôme en la taille du problème (L).

La réponse est venue de Khachiyan (1979) lorsqu'il a montré que la méthode des ellipsoïdes était de complexité globale $O(n^4L)$ itérations pour la programmation linéaire [45]. Cependant, et tandis que la méthode du simplexe de complexité exponentielle, fonctionnait bien en pratique, la méthode des ellipsoïdes s'est avérée nettement moins efficace que la méthode du simplexe. Il restait donc de trouver des méthodes polynômiales qui soient efficaces en pratique.

En 1984, Karmarkar proposa une méthode projective de points intérieurs à convergence polynômiale $O(n^{3.5}L)$ itérations concurrente à la méthode de simplexe pour résoudre des problèmes d'optimisation linéaire de grande taille [42]. Depuis lors, plusieurs méthodes de points intérieurs ont été proposées et étudiées.

Dans les années 90, les méthodes de trajectoire centrale pour la programmation linéaire sont avérées être très efficaces en pratique, en 1989 Kojima, Mizuno et Yoshise [49] ont proposé un algorithme de complexité polynômiale $O(nL)$ itérations, en 1990 Mehrotra [53] a décrit un autre algorithme de trajectoire centrale prédicteur-correcteur très efficace en pratique de complexité polynômiale $O(pnL)$ itérations.

Ces progrès rivalisant, n'échappent pas tout de même aux inconvénients théoriques, algorithmiques et numériques, entre autre, le problème d'initialisation qui devient de plus en plus difficile en passant aux autres problèmes mathématiques (comment choisir un point initial réalisable près de la trajectoire centrale pour démarrer l'algorithme ?). Ce problème d'initialisation est au centre de notre étude, il figure depuis des années au menu de la recherche dans le domaine de la programmation mathématique.

Beaucoup de travaux sont faites pour atténuer à cet inconvénient à savoir les *MPIs* avec poids et les *MPIs* non réalisables, pour plus de détails voir les réf-

rences [4, 26, 43, 65, 67].

Récemment, une alternative particulièrement intéressante pour nous est étudiée, celle des *MPIs* basées sur les fonctions barrières qui sont définies par une grande classe de fonctions uni-variées appelées fonctions noyaux qui ont été utilisés avec succès pour concevoir de nouvelles *MPIs* pour l'optimisation linéaire. Les fonctions noyaux jouent un rôle important dans la conception et l'analyse d'algorithmes de points intérieurs primaux-duaux. Elles ne sont pas seulement utilisées pour déterminer les directions de recherche mais aussi pour mesurer la distance entre l'itération donnée et le centre des algorithmes. En particulier, Peng et al. [61] ont introduit une nouvelle classe de fonction noyau dite auto-régulières et conçue des *MPIs* primales-duales basées sur des proximités auto-régulières pour l'optimisation linéaire. Ils ont amélioré la complexité des *MPIs* de $O(n \log \frac{n}{\epsilon})$ à $O(\sqrt{n} \log n \log \frac{n}{\epsilon})$. Plus tard, Bai et al. [7] ont présenté une grande classe de fonctions noyaux éligibles, qui est assez générale et comprend la fonction logarithmique classique et les fonctions auto-régulières, ainsi que de nombreuses fonctions non auto-régulières comme cas spéciaux. La complexité obtenue pour l'optimisation linéaire est aussi bonne que celle de [61]. En 2012, El Ghami et al. [29] ont introduit une fonction noyau trigonométrique pour l'optimisation linéaire. Ces méthodes ont été étudiées par plusieurs chercheurs avec d'autres fonctions noyaux dans leurs articles [2, 7, 14, 22, 25, 29, 46, 51, 60, 59, 61, 63]. Ceci a permis d'améliorer tous les résultats théoriques obtenus surtout pour les méthodes à long-pas qui sont en pratique beaucoup plus efficaces que les méthodes à court-pas [6].

La programmation quadratique convexe (*PQC*) a reçu une attention considérable de la part des chercheurs en raison du lien étroit entre la programmation linéaire, la programmation quadratique convexe et la programmation non linéaire convexe. Monteiro et Adler ont proposé un algorithme de points intérieurs de type trajectoire centrale primal-dual pour *PQC* dans [58] qui nécessite un total de $O(\sqrt{n}L)$ d'itérations. Yu et al. [83] ont développé un algorithme polynomial prédicteur correcteur de points intérieurs pour *PQC* à complexité $O(\sqrt{n}L)$ d'itérations. Achache [1] a fourni une nouvelle méthode primale-duale de type trajectoire centrale pour *PQC* et a prouvé que l'algorithme à court-pas nécessite $O(2\sqrt{n} \log \frac{n}{\epsilon})$ d'itérations.

Roumili [66] a proposé une *MPI* avec poids pour résoudre un *PQC*. Lu et al. [52], Roumili et Boudjellal [68], Roumili et al. [4] ont présenté des algorithmes de points intérieurs non réalisable pour *PQC*. Wang et al. [77] ont proposé une classe d'algorithmes de points intérieurs primaux-duaux basés sur une fonction noyau et ont obtenu la complexité la plus connue actuellement pour la méthode à grand pas, à savoir $O(\sqrt{n} \log n \log \frac{n}{\varepsilon})$ itérations. Pour certaines autres méthodes connexes pour *PQC*, nous nous référons à [78, 80].

Dans cette thèse, Nous nous intéressons aux problèmes de la programmation quadratique convexe (*PQC*) où, nous avons proposé de nouveaux algorithmes de points intérieurs primaux-duaux basés sur des nouvelles fonctions noyaux. La première fonction noyau avec terme barrière exponentiel proposée généralise celle donnée par Bai et al. [7] lorsque $p = 1$. Cette fonction possède des propriétés faciles à vérifier. Ces propriétés nous permettent d'améliorer le comportement numérique de l'algorithme proposé. Des tests numériques ont été effectués pour étudier le comportement de l'algorithme avec différents paramètres p et θ , où p est le degré de barrière de la fonction du noyau et θ est le paramètre barrière de mise à jour. Ce travail a été publié dans une revue spécialisée « optimization journal » [18]. La deuxième fonction noyau paramétrée améliore de manière significative les résultats théoriques et pratiques obtenus par Peng et al. dans [60] et Bai et al. dans [9], la complexité polynomiale obtenue liée d'une *MPI* à grand pas à $O(\sqrt{n} \log n \log \frac{n}{\varepsilon})$ lorsque $p = \frac{\log n}{2} - 2$, qui est actuellement la meilleure complexité polynomiale. Ce travail est accepté pour être publié dans une revue spécialisée « Boletim da Sociedade Paranaense de Matemática (BSPM) journal » [19].

La thèse est composée de trois chapitres organisés comme suit :

- Le premier chapitre comprend les outils de l'analyse convexe et l'optimisation convexe nécessaires pour une bonne compréhension de la suite.
- Le second chapitre est consacré à la méthode de points intérieurs (*MPIs*) primale-duale de type trajectoire centrale basée sur une fonction noyau pour la programmation quadratique convexe (*PQC*). En premier, nous rappelons les notions de base des fonctions noyaux. Ensuite, nous présentons le principe, l'algorithme, l'analyse de complexité de la méthode de trajectoire cen-

trale basée sur les fonctions noyaux qualifiées en général.

- Dans le troisième chapitre, on étudie théoriquement et numériquement les *MPIs* basées sur deux nouvelles fonctions noyaux citées auparavant pour *PQC* ; à savoir : l'une est du type exponentiel et l'autre polynomial (de Laurent). Enfin, des expérimentations numériques comparatives sont présentées pour montrer leur efficacité.
- On terminera cette thèse par une conclusion générale et perspectives.

NOTATIONS PRINCIPALES

x est un k – vecteur	$x \in \mathfrak{R}^k$;
$\ \cdot\ $	la norme euclidienne ;
$C^1(D)$	l'ensemble des fonctions continûments différentiables sur D ;
∇f	le gradient de la fonction f ;
$C^2(D)$	l'ensemble des fonctions deux fois continûments différentiables sur D ;
$\nabla^2 f(x)$	le hessien de la fonction f ;
A^t	le transposé de A ;
$B_\varepsilon(x^*)$	$\{x \in \mathfrak{R}^n : \ x - x^*\ < \varepsilon\}$ est une boule ouverte de centre x^* et de rayon ε ;
$int(D)$	l'intérieur du domaine D ;
\mathfrak{R}_{++}^m	$\{x \in \mathfrak{R}^m : x > 0\}$;
\mathfrak{R}_+^m	$\{x \in \mathfrak{R}^m : x \geq 0\}$;
$\ker(A)$	$\{x \in \mathfrak{R}^n : Ax = 0\}$;
$C^\infty(\mathfrak{R}_{++}^n)$	l'ensemble des fonctions infiniments continûments différentiables sur \mathfrak{R}_{++}^n ;
A^{-1}	la matrice inverse de A ;
X^{-1}	$diag\left(\frac{1}{x_1}, \dots, \frac{1}{x_n}\right)$;
X^{-2}	$diag\left(\frac{1}{x_1^2}, \dots, \frac{1}{x_n^2}\right)$;
$C^3(D)$	l'ensemble des fonctions trois fois continûments différentiables sur D ;

v^{-2}	$\left(\frac{1}{v_1^2}, \dots, \frac{1}{v_n^2}\right)^t$;
xz	$(x_1z_1, \dots, x_nz_n)^t$;
$\frac{x}{z}$	$\left(\frac{x_1}{z_1}, \dots, \frac{x_n}{z_n}\right)^t$;
e	$(1, \dots, 1)^t$;
$\langle \cdot, \cdot \rangle$	le produit scalaire ;
$ \cdot $	la valeur absolue ;

Terminologie :

<i>MPIs</i>	Méthodes de points intérieurs ;
<i>PQC</i>	Programmation quadratique convexe ;
<i>PM</i>	Le programme mathématique ;
<i>KKT</i>	Karush-Kuhn-Tucker ;
<i>DM</i>	Le dual d'un programme mathématique ;
<i>PQ</i>	Le programme quadratique ;
<i>DQC</i>	Le dual d'un programme quadratique convexe ;
<i>OL</i>	Optimisation linéaire ;

CHAPITRE 1

NOTIONS D'ANALYSE CONVEXE ET OPTIMISATION QUADRATIQUE

Pour rendre cette thèse facile à lire, nous allons introduire certaines notions de base de l'analyse convexe. Ces notions sont utiles pour les prochains chapitres.

Après un bref rappel de la programmation mathématique, nous commençons par la position du problème et sa classification, puis nous introduisons les principaux résultats d'existence et d'unicité d'un optimum. Nous rappelons ensuite le principe de la qualification des contraintes, ainsi que les conditions d'optimalité en détail, notamment la dualité au sens de Lagrange.

Puis, nous passons à la programmation quadratique convexe qui fait l'objet de cette thèse. Enfin, nous donnons un aperçu sur les différents types des méthodes de résolution d'un programme quadratique convexe.

1.1 Notions d'analyse convexe

La convexité joue un rôle important dans la théorie de l'optimisation mathématique, un problème convexe possède des propriétés remarquables qui permettent d'en faciliter la résolution. La notion de convexité prend deux formes : un ensemble

convexe et une fonction convexe.

1.1.1 Ensembles et fonctions convexes

Définition 1.1.1. Un ensemble D de \mathbb{R}^n est dit

— un ensemble convexe si

$$\lambda x + (1 - \lambda)y \in D, \forall x, y \in D, \forall \lambda \in [0, 1].$$

— un ensemble affine si

$$\lambda x + (1 - \lambda)y \in D, \forall x, y \in D, \forall \lambda \in \mathbb{R}.$$

— un polyèdre convexe s'il est de la forme suivante

$$D = \{x \in \mathbb{R}^n : Ax \leq b\},$$

où $A \in \mathbb{R}^{m \times n}$ et b est un m -vecteur.

Définition 1.1.2. Soit $f : D \rightarrow \mathbb{R}$ une fonction et D un ensemble convexe de \mathbb{R}^n , alors

— f est dite convexe sur D si

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \forall x, y \in D, \forall \lambda \in [0, 1].$$

— f est dite strictement convexe sur D si

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \forall x, y \in D \text{ et } x \neq y, \forall \lambda \in]0, 1[.$$

— f est dite affine sur D si

$$f(\lambda x + (1 - \lambda)y) = \lambda f(x) + (1 - \lambda)f(y), \forall x, y \in D, \forall \lambda \in \mathbb{R}.$$

— f est dite coercive sur D si $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$.

1.1.2 Convexité et différentiabilité

Lemme 1.1.3. ([23]) Soit $f \in C^1(\Omega)$ où Ω est un ouvert et $D \subset \Omega$ un ensemble convexe, alors les propriétés suivantes sont équivalentes :

- (i) f est une fonction convexe.
- (ii) $f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle, \forall x, y \in D.$
- (iii) $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0, \forall x, y \in D.$

De plus, f est dite strictement convexe si les inégalités (ii) et (iii) du lemme précédent rendues strictes pour $x \neq y$.

Définition 1.1.4. Soit $f \in C^2(\Omega)$ où Ω est un ouvert et $D \subset \Omega$ un ensemble convexe, on dit que :

- f est convexe si et seulement si la matrice hessienne $\nabla^2 f(x)$ de f est une matrice semi-définie positive c'est à dire que $y^t \nabla^2 f(x) y \geq 0, \forall x, y \in D$ (ou toutes les valeurs propres de $\nabla^2 f(x)$ sont positives).
- f est strictement convexe si et seulement si la matrice hessienne $\nabla^2 f(x)$ de f est une matrice définie positive c'est à dire que $y^t \nabla^2 f(x) y > 0, \forall x, y \in D, y \neq 0$ (ou toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives).

1.1.3 Notations asymptotiques et vitesse de convergence

Dans ce paragraphe, nous donnons la définition des notations O et Θ qu'on va utiliser dans cette thèse. Puis, nous présentons les types de vitesse de convergence d'un algorithme.

Définition 1.1.5. Soient $h, g : \mathbb{N} \rightarrow \mathbb{R}_+$ deux fonctions, on note

- $h(n) = O(g(n))$ lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

$$h(n) \leq cg(n).$$

- $h(n) = \Theta(g(n))$ lorsqu'il existe des entiers c_1, c_2 et n_0 tels que pour tout $n \geq n_0$,

$$c_1g(n) \leq h(n) \leq c_2g(n).$$

Définition 1.1.6. ([37]) Soit $(x^k)_{k \in \mathbb{N}}$ une suite engendrée par un algorithme qui converge vers x^* et soit $e_k = x^k - x^*$ l'erreur commise à chaque itération, on suppose que la suite est non stationnaire, c.à.d. $x^k \neq x^*$ pour tout $k \in \mathbb{N}$, et que

$$\lim_{k \rightarrow +\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = \beta \in [0, 1].$$

On s'intéresse à la "vitesse de convergence" de la suite $(x^k)_{k \in \mathbb{N}}$. On dit que

- La convergence est linéaire si $\beta \in]0, 1[$.
- La convergence est super linéaire si $\beta = 0$.
- La convergence est d'ordre q si

$$\lim_{k \rightarrow +\infty} \frac{\|e_{k+1}\|}{\|e_k\|^q} = \nu > 0.$$

Dans le cas $q = 2$ la convergence est dite quadratique.

1.2 Programmation mathématique

La programmation mathématique est l'outil le plus utilisé pour résoudre analytiquement ou numériquement les problèmes à raison de ses nombreuses applications. Un problème de programmation mathématique consiste à minimiser ou maximiser une fonction f sur un domaine D .

1.2.1 Position du problème et sa classification

Un programme mathématique est un problème défini en général comme suit :

$$\begin{cases} \min f(x), \\ x \in D, \end{cases} \quad (PM)$$

où $D = \{x \in \mathbb{R}^n : h_i(x) \leq 0, i = 1, \dots, m, g_j(x) = 0, j = 1, \dots, p\}$, $h_i, g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ et $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$.

On appelle f la fonction objectif et D l'ensemble des contraintes ou l'ensemble des solutions réalisables.

Définition 1.2.1. — On appelle solution réalisable de (PM) tout point x vérifiant les contraintes i.e., $x \in D$.

— Une solution réalisable x^* est dite solution optimale globale de (PM) si

$$f(x) \geq f(x^*), \forall x \in D.$$

— Une solution réalisable x^* est dite solution optimale locale de (PM) s'il existe $\varepsilon > 0$:

$$f(x) \geq f(x^*), \forall x \in D \cap B_\varepsilon(x^*).$$

Le problème (PM) est classé selon deux notions qui jouent un rôle majeur à savoir la convexité et la différentiabilité des fonctions du problème :

- Un problème convexe : si f, h_i sont des fonctions convexes et g_j sont des fonctions affines.
- Un problème différentiable : si f, h_i, g_j sont des fonctions différentiables.

Nous donnons dans le paragraphe ci-dessous les fondamentaux résultats d'existence et d'unicité d'une solution optimale de (PM).

1.2.2 Principaux résultats d'existence et d'unicité d'un optimum

Théorème 1.2.2. ([12]) (Weierstrass) Soit D un compact (fermé et borné) non vide de \mathbb{R}^n , si f est continue sur D alors (PM) admet au moins une solution optimale globale $x^* \in D$.

Corollaire 1.2.3. ([12]) Soit D un fermé non vide de \mathbb{R}^n , si f est continue et coercive sur D alors (PM) admet au moins une solution optimale globale.

Théorème 1.2.4. ([12]) Soit D un convexe non vide de \mathbb{R}^n , si f est strictement convexe sur D alors la solution optimale de (PM), si elle existe, est unique.

1.2.3 Qualification des contraintes

Définition 1.2.5. La contrainte $h_i(x) \leq 0$ est dite active ou saturée en $\bar{x} \in D$ si $h_i(\bar{x}) = 0$.

La contrainte d'égalité $g_j(x)$ est saturée par définition.

Voici les trois critères courants de qualification :

- Le premier critère est donné par Karlin [41] en 1959 :
les contraintes sont qualifiées en tout point de D si D est un polyèdre convexe i.e., h_i, g_j sont affines.
- Le deuxième critère est donné par Slater [72] en 1950 :
les contraintes sont qualifiées en tout point de D si D est convexe (i.e., h_i sont convexe et g_j sont affines) et $\text{int}(D) \neq \emptyset$.
- Le dernier critère est donné par Fiacco et McCormick [31] en 1968 :
les contraintes sont qualifiées en $\bar{x} \in D$ si les gradients de toutes les fonctions des contraintes saturées en \bar{x} sont linéairement indépendants.

Définition 1.2.6. *Un point $\bar{x} \in D$ est dit régulier si l'une des conditions de qualifications précédentes est satisfaite en \bar{x} .*

1.2.4 Conditions d'optimalité

Ce paragraphe est destiné à rappeler les conditions nécessaires d'optimalité du premier ordre (Karush-Kuhn-Tucker) qui sont l'origine de certaines méthodes de résolution d'un (PM), telle que la méthode de Lemke (voir [20]).

Reprenons le programme mathématique suivant :

$$\left\{ \begin{array}{l} \min f(x), \\ h_i(x) \leq 0, \quad i = 1, \dots, m, \\ g_j(x) = 0, \quad j = 1, \dots, p, \\ x \in \mathbb{R}^n. \end{array} \right. \quad (PM)$$

Le lagrangien associé au problème (PM) est défini par :

$$L(x, \lambda, \eta) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^p \eta_j g_j(x), \quad (1.2.1)$$

où λ est un m -vecteur et η est un p -vecteur.

Théorème 1.2.7. ([56]) (Karush-Kuhn-Tucker) *Soit x^* un minimum local régulier du*

problème (PM). Alors, il existe les multiplicateurs $\lambda \in \mathbb{R}^m$ et $\eta \in \mathbb{R}^p$ tels que :

$$\left\{ \begin{array}{l} (a) \quad \nabla_x L(x^*, \lambda, \eta) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla h_i(x^*) + \sum_{j=1}^p \eta_j \nabla g_j(x^*) = 0, \\ (b) \quad \lambda_i h_i(x^*) = 0, \quad i = 1, \dots, m, \\ (c) \quad g_j(x^*) = 0, \quad j = 1, \dots, p, \\ (d) \quad \lambda_i \geq 0, \quad i = 1, \dots, m. \end{array} \right. \quad (\text{KKT})$$

Ces conditions sont appelées conditions d'optimalité de Karush-Kuhn-Tucker (KKT). Les conditions (a) et (b) s'appellent les conditions d'optimalité et de complémentarité respectivement. Dans la condition (c), on reconnaît la condition d'admissibilité des contraintes d'égalité en x^* . Le vecteur λ dans la condition (d) est appelé multiplicateur de Lagrange associé aux contraintes d'inégalité à la solution x^* .

Remarque 1.2.8. Dans le cas (PM) est convexe, un point x^* est un minimum global du problème (PM) si et seulement s'il satisfait les conditions (KKT) de Karush-Kuhn-Tucker.

Théorème 1.2.9. ([56]) Si f est une fonction convexe continûment différentiable, une condition nécessaire et suffisante pour que x^* soit un optimum global de f est que $\nabla f(x^*) = 0$.

1.2.5 Dualité au sens de Lagrange

Nous présentons dans ce paragraphe le problème dual (DM) d'un problème mathématique (PM) convexe au sens de Lagrange comme suit :

$$\left\{ \begin{array}{l} \max_{(\lambda, \eta)} \min_{x \in D} L(x, \lambda, \eta), \\ \nabla_x L(x, \lambda, \eta) = 0, \\ \lambda \in \mathbb{R}_{++}^m, \eta \in \mathbb{R}^p. \end{array} \right. \quad (\text{DM})$$

Le lagrangien associé au problème (PM) est écrit dans (1.2.1), donc (DM) devient :

$$\left\{ \begin{array}{l} \max_{(\lambda, \eta)} \min_{x \in D} \left[f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^p \eta_j g_j(x) \right], \\ \nabla f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x) + \sum_{j=1}^p \eta_j \nabla g_j(x) = 0, \\ \lambda \in \mathbb{R}_{++}^m, \eta \in \mathbb{R}^p. \end{array} \right.$$

1.3 Programmation quadratique convexe

La programmation quadratique est une branche de l'optimisation non linéaire où la fonction objectif à minimiser est une fonction quadratique et les contraintes définissant le domaine des solutions réalisables sont linéaires et/ou quadratiques.

Son importance réside dans ses propriétés théoriques, ses applications dans différents domaines scientifiques et plusieurs disciplines telles que : la finance, la médecine, l'économie et la télécommunication.

En fait, plusieurs problèmes réels et académiques peuvent être modélisés sous forme de programme quadratique par exemple : l'optimisation de portefeuille en finance, transmission de faisceaux multi-couches, problèmes dans les faisceaux multi-capteurs et des problèmes géométriques.

Un programme quadratique sous contraintes linéaires peut s'écrire sous la forme suivante :

$$\left\{ \begin{array}{l} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ Dx \leq e, \\ x \in \mathbb{R}^n, \end{array} \right. \quad (PQ)$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique, $A \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{p \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et $e \in \mathbb{R}^p$.

Tout programme quadratique peut se ramener à l'une des formes suivantes :

◆ **Forme standard (FS)**

$$\left\{ \begin{array}{l} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ x \geq 0. \end{array} \right.$$

◆ **Forme canonique (FC)**

$$\left\{ \begin{array}{l} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax \leq b, \\ x \geq 0. \end{array} \right.$$

En raison de nombreuses applications d'un programme quadratique, plusieurs variantes de ce problème ont été traitées par plusieurs chercheurs :

cas convexe, cas non convexe, avec variables entières ou variables binaires, on cite à titre d'exemples les travaux [33, 35, 50].

La notion de convexité joue un rôle majeur dans la programmation mathématique, un problème convexe possède des propriétés remarquables permettant d'en faciliter la résolution. Rockafellar dans [64] a montré qu'une distinction majeure doit être faite entre les problèmes convexes et les problèmes non convexes. Dans cette thèse, nous nous intéressons aux problèmes quadratiques convexes sous forme standard.

1.3.1 Problème quadratique convexe primal

Un programme quadratique convexe primal sous forme standard s'écrit comme suit :

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ x \geq 0, \end{cases} \quad (PQC)$$

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

On note par

$$\begin{aligned} F_{(PQC)} &= \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}, \\ F_{(PQC)}^0 &= \{x \in \mathbb{R}^n : Ax = b, x > 0\}, \end{aligned}$$

les ensembles des solutions réalisables et strictement réalisables de (PQC) , respectivement.

1.3.2 Problème quadratique convexe dual

Le problème dual (DQC) associé au problème primal (PQC) peut s'écrire sous la forme suivante :

$$\begin{cases} \max -\frac{1}{2}x^t Qx + b^t y, \\ A^t y + z - Qx = c, \\ z \geq 0, x \geq 0. \end{cases} \quad (DQC)$$

où $y \in \mathbb{R}^m, z \in \mathbb{R}^n$.

La dualité joue un rôle fondamental dans l'étude et la résolution de (PQC). Entre autre, elle fournit des informations très utiles. On note par

$$\begin{aligned} F_{(DQC)} &= \left\{ (y, z) \in \mathbb{R}^m \times \mathbb{R}^n : A^t y + z - Qx = c, z \geq 0 \right\}, \\ F_{(DQC)}^0 &= \left\{ (y, z) \in \mathbb{R}^m \times \mathbb{R}^n : A^t y + z - Qx = c, z > 0 \right\}, \end{aligned}$$

les ensembles des solutions réalisables et strictement réalisables de (DQC), respectivement, et

$$F = F_{(PQC)} \times F_{(DQC)} \text{ et } F^0 = F_{(PQC)}^0 \times F_{(DQC)}^0,$$

les ensembles des solutions réalisables primales-duales et strictement réalisables primales-duales de (PQC) et (DQC) respectivement.

Au long de cette thèse, on suppose que :

(H1) $F^0 \neq \emptyset$ c.-à-d. un point primal-dual strictement réalisable existe.

(H2) $\text{rang}(A) = m < n$ c.-à-d. A est une matrice de rang plein.

1.3.3 Dualité en programmation quadratique

Théorème 1.3.1. ([78])(Dualité faible) Si $x \in F_{(PQC)}$ et $(y, z) \in F_{(DQC)}$ alors

$$\frac{1}{2} x^t Q x + c^t x \geq -\frac{1}{2} x^t Q x + b^t y.$$

Théorème 1.3.2. ([78])(Dualité forte) Soient $x^* \in F_{(PQC)}$ et $(y^*, z^*) \in F_{(DQC)}$ tels que $\frac{1}{2} (x^*)^t Q x^* + c^t x^* = -\frac{1}{2} (x^*)^t Q x^* + b^t y^*$, alors x^* et (y^*, z^*) sont des solutions optimales pour (PQC) et (DQC) respectivement.

Définition 1.3.3. (Saut de dualité) Soit $(x, y, z) \in F$, la quantité

$$\left(\frac{1}{2} x^t Q x + c^t x \right) - \left(-\frac{1}{2} x^t Q x + b^t y \right) = x^t z,$$

est appelée saut de dualité des deux problèmes (PQC) et (DQC) en (x, y, z) .

Dans le théorème suivant, on peut exprimer la condition pour x^* et (y^*, z^*) d'être des solutions optimales de (PQC) et (DQC) respectivement sous la forme d'une condition dite de complémentarité.

Théorème 1.3.4. ([78])(Des écarts complémentaires) Si $x^* \in F_{(PQC)}$ et $(y^*, z^*) \in F_{(DQC)}$ alors x^* et (y^*, z^*) sont des solutions optimales de (PQC) et (DQC) respectivement si et seulement si

$$(x^*)^t z^* = 0 \iff x^* z^* = 0 \iff x_i^* z_i^* = 0, \forall i = 1, \dots, n.$$

Remarque 1.3.5. L'existence de solution optimale de l'un des problèmes (PQC) et (DQC) implique l'optimalité de l'autre et leurs valeurs optimales sont égales.

Si l'un des deux problèmes (PQC) et (DQC) a une valeur optimale non finie, l'autre n'a pas de solution.

1.3.4 Méthodes de résolution d'un (PQC)

Il existe plusieurs méthodes pour la résolution des problèmes quadratiques avec contraintes linéaires. On peut les classer en trois grandes catégories :

1. Les méthodes de type gradient : nous citons par exemple la méthode du gradient conjugué et la méthode du gradient projeté.
2. Les méthodes simpliciales : nous citons par autre la méthode du gradient réduit (Wolfe).
3. Les méthodes de points intérieurs : nous citons les méthodes affines, les méthodes projectives, les méthodes de réduction du potentiel et les méthodes de trajectoire centrale.

Nous donnons par la suite le principe, l'algorithme et les résultats de convergence pour chaque catégorie de méthodes.

Les méthodes de type gradient

1. Méthode du gradient conjugué

La méthode du gradient conjugué a été proposée en premier par Hestenes [36] en 1952 pour résoudre les systèmes des équations linéaires. Puis en

1991, Fletcher et Reeves [32] l'ont généralisé pour résoudre les problèmes d'optimisation non linéaire.

Cette méthode s'applique directement au problème quadratique convexe sans contraintes, elle a été adaptée par plusieurs chercheurs [55] pour résoudre un problème quadratique convexe sous contraintes linéaires (PQC) par le changement de variable suivant :

$$x = x_0 + P_A y \quad \text{tel que : } Ax_0 = b, P_A = I - A^t (AA^t)^{-1} A \text{ et } y \in \mathbb{R}^n,$$

où P_A c'est l'opérateur de projection sur $\ker(A)$.

Ce changement permet de ramener le problème (PQC) à un problème quadratique sans contraintes s'écrivant comme suit

$$\begin{cases} \min \phi(y) = \frac{1}{2} (x_0 + P_A y)^t Q (x_0 + P_A y) + c^t (x_0 + P_A y), \\ y \in \mathbb{R}^n, \end{cases}$$

où $P_A y = (I - A^t (AA^t)^{-1} A) y$.

Bien évidemment, si y^* est un minimum de $\phi(y)$ son correspondant $x^* = x_0 + P_A y^*$ est une solution optimale pour (PQC).

Principe de la méthode

Elle consiste à construire progressivement des directions d_0, d_1, \dots, d_k mutuellement conjuguées par rapport à la matrice $\bar{Q} = (P_A)^t Q P_A$.

A chaque étape k , la direction d_k est obtenue par combinaison linéaire du gradient $-\nabla \phi(y^k)$ en y^k et les directions précédentes, d_0, d_1, \dots, d_{k-1} , les coefficients de la combinaison linéaire étant choisis de telle sorte que d_k soit conjuguée par rapport à toutes les directions précédentes.

L'algorithme de cette méthode est décrit comme suit :

Algorithme 1 : Algorithme du gradient conjugué

Initialisation : Soit y^0 le point de départ ;

Une tolérance choisie $\varepsilon > 0$;

Poser $g_0 = \nabla\phi(y^0)$, $d_0 = -g_0$, $k = 0$;

Tant que $\|\nabla\phi(y^k)\| \geq \varepsilon$ **faire**

$$\lambda_k = -\frac{g_k^t d_k}{d_k^t Q d_k}.$$

$$y^{k+1} = y^k + \lambda_k d_k.$$

Puis

$$d_{k+1} = -g_{k+1} + \beta_k d_k.$$

Avec

$$\beta_k = \frac{g_{k+1}^t \bar{Q} d_k}{d_k^t \bar{Q} d_k}; k = k + 1.$$

Fin Tant que

Pour démontrer la validité de l'algorithme (convergence en plus n étapes), il suffit donc de vérifier que les directions engendrées par l'**Algorithme 1** ($d_k, k = 0, \dots, n - 1$) sont mutuellement conjuguées, pour cela nous décrirons le théorème suivant :

Théorème 1.3.6. ([56]) *A une itération k quelconque de l'algorithme 1 où l'optimum de f n'est pas encore atteint (c'est-à-dire $g_i \neq 0, i = 0, 1, \dots, k$) on a :*

(i) $\lambda_k = -\frac{g_k^t d_k}{d_k^t Q d_k} \neq 0.$

(ii) $\beta_k = \frac{g_{k+1}^t (g_{k+1} - g_k)}{g_k^t g_k} = \frac{g_{k+1}^t g_{k+1}}{g_k^t g_k}.$

(iii) *Les directions d_0, d_1, \dots, d_{k-1} engendrées par l'algorithme sont mutuellement conjuguées.*

2. Méthode du gradient projeté

La technique de base utilisée pour cette méthode est de projeter le gradient sur la frontière du domaine réalisable. Ceci donne un cheminement le long de la frontière dans la direction de la plus forte pente autorisé par les contraintes. En 1960, Rosen est le premier qui introduit ce principe dans son algorithme du gradient projeté. La méthode du gradient projeté est essentiellement intéressante dans le cas des contraintes linéaires.

Supposons donc le problème (PQ) donné sous la forme

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ A_i x = b_i, i \in I_1 = \{1, \dots, m\}, \\ D_i x \leq e_i, i \in I_2 = \{1, \dots, p\}, \\ x \in \mathbb{R}^n, \end{cases}$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique, $A_i \in \mathbb{R}^{1 \times n}$, $D_i \in \mathbb{R}^{1 \times n}$, $c \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ et $e_i \in \mathbb{R}$.

Soit $I^0(x) = \{i : D_i x = e_i\} \cup I_1$ l'ensemble des indices des contraintes saturées au point x .

Principe de la méthode

On va chercher une direction de déplacement d ($\|d\| = 1$) au point x qui permet de diminuer le plus possible la fonction objectif $f(x)$ (donc qui rend $(\nabla f(x))^t d$ minimal), mais qui, au moins pour un petit déplacement, permette de rester dans l'ensemble des solutions réalisables.

Nous notons par

- A_0 la sous-matrice de A constituée par les lignes $i \in I^0(x)$ de A et D .
- $\text{rang}(A) = |I^0(x)| = q$, nous supposons que A_0 est de rang plein pour éviter le problème de dégénérescence au point x^0 .
- S^0 l'ensemble des vecteurs (sous espace vectoriel) est défini par

$$S^0 = \{y : A_0 y = 0\}.$$

Il s'agit donc de rechercher une direction $d \in S^0$ tel que $(\nabla f(x))^t d$ soit minimal avec $\|d\| = 1$ (la contrainte de normalisation).

Le théorème suivant donne la solution explicite de ce problème par la projection du vecteur $-\nabla f(x)$ sur S^0 .

Théorème 1.3.7. ([56]) Si A_0 est une matrice $q \times n$ de rang plein $q \leq n$, la

solution optimale du problème :

$$\begin{cases} \min (\nabla f(x))^t d, \\ A_0 d = 0, \\ \|d\| = 1, \end{cases}$$

est $d = \bar{y}/\|\bar{y}\|$, où \bar{y} est la projection de $-\nabla f(x)$ sur S^0 , donnée par :

$$\bar{y} = -P_0 \nabla f(x) = -\left(I - A_0^t (A_0 A_0^t)^{-1} A_0\right) \nabla f(x),$$

où $P_0 = I - A_0^t (A_0 A_0^t)^{-1} A_0$ désigne la matrice de projection sur S^0 .

Description

(a) Si $\bar{y} = -P_0 \nabla f(x) \neq 0$ alors \bar{y} est une direction de descente car

$$(\nabla f(x))^t \bar{y} = -\bar{y}^t (\bar{y} + \bar{z}) = -\bar{y}^t \bar{y} = -\|\bar{y}\|^2 < 0 \text{ tel que : } \bar{z} = (I - P_0) \nabla f(x).$$

- Après l'obtention de la direction de déplacement, on détermine le pas de déplacement maximal α_{\max} autorisé par les contraintes tel que :

$$\alpha_{\max} = \max \{ \alpha : \alpha \geq 0; x + \alpha \bar{y} \in \text{Dom} \},$$

où $\text{Dom} = \{x \in \mathbb{R}^n : A_i x = b_i, \forall i \in I_1; D_i x \leq e_i, \forall i \in I_2\}$.

- Le nouveau itéré x' est un point minimisant la fonction $f(x + \alpha \bar{y})$ sur $[0, \alpha_{\max}]$.
- Pour calculer la nouvelle direction de déplacement $\bar{y}' = -P_0 \nabla f(x')$, il doit déterminer le nouvel ensemble $I^0(x')$ et la nouvelle matrice de projection P_0 .

(b) Voyons maintenant ce qui se passe lorsque $\bar{y} = -P_0 \nabla f(x) = 0$:

Donc, on a :

$$\nabla f(x) + A_0^t u = 0 \text{ avec } u = -\left(A_0 A_0^t\right)^{-1} A_0 \nabla f(x). \quad (1.3.1)$$

- Si $u \geq 0$, l'équation (1.3.1) représente les conditions de (KKT) au point x donc on peut dire que x est un optimum local du problème initial.
- Sinon, le point x ne vérifie pas les conditions nécessaires d'optimalité et il faut déterminer une autre direction de déplacement. Pour cela, on supprime dans $I^0(x)$ l'une des contraintes i pour lesquelles $u_i < 0$. On trouve une nouvelle matrice A'_0 et une nouvelle matrice de projection P'_0 pour calculer la nouvelle direction de déplacement $\bar{y}' = -P'_0 \nabla f(x)$.

Remarque 1.3.8. *Les ensembles des contraintes saturées entre l'itération k et l'itération $k + 1$ ($I^0(x^k)$ et $I^0(x^{k+1})$) ne diffèrent généralement que par un élément ou plus. Par conséquent, les matrices A_0 ne diffèrent que par une ligne et la nouvelle matrice de projection peut être calculée directement à partir de l'ancienne, ce qui peut réduire sensiblement les calculs nécessaires à chaque itération. Donc, la méthode du gradient projeté a un caractère intéressant par rapport à les autres méthodes de type gradient.*

L'algorithme du gradient projeté est donc donné comme suit :

Algorithme 2 : Algorithme du gradient projeté.

Initialisation :

Soit $x^0 \in Dom$ le point de départ ;

Une tolérance choisie $\varepsilon > 0$; $k = 0$;

Tant que $(\|y\| > \varepsilon \text{ et } u_{\min} = \min_{j=1,\dots,n} u_j < -\varepsilon)$ **faire**

Soit A_0^k la matrice dont les lignes correspondent aux contraintes saturées au point x^k .

Calculer

$$P_0 = I - \left(A_0^k \right)^t \left(A_0^k \left(A_0^k \right)^t \right)^{-1} A_0^k,$$

$$\bar{y} = -P_0^k \nabla f \left(x^k \right),$$

$$u = - \left(A_0^k \left(A_0^k \right)^t \right)^{-1} A_0^k \nabla f \left(x^k \right),$$

$$u_{\min} = \min_{j=1,\dots,n} u_j.$$

Si $(\bar{y} = 0 \text{ et } u_{\min} < 0)$ **alors**

Soit A_0^k la matrice obtenue à partir de A_0^k en supprimant la ligne d'indice i , P_0' la nouvelle matrice de projection correspondante et $\bar{y}' = -P_0' \nabla f (x)$.

$$\bar{y} = \bar{y}' ;$$

Sinon

Si $(\bar{y} = 0 \text{ et } u_{\min} \geq 0)$ **alors**

Le point x^k satisfait les conditions nécessaires de (KKT).

Fin des itérations (sortie de la boucle).

Fin Si

Fin Si

Déterminer le pas de déplacement maximal α_{\max} qui définit par :

$$\alpha_{\max} = \max \{ \alpha : \alpha \geq 0 ; x + \alpha \bar{y} \in Dom \} ;$$

$$\text{Déterminer } x^{k+1} \text{ tel que : } f \left(x^{k+1} \right) = \min_{0 \leq \alpha \leq \alpha_{\max}} f \left(x^k + \alpha \bar{y} \right) ;$$

$$k \leftarrow k + 1 ;$$

Fin Tant que

Pour plus de détails sur les méthodes de type gradient nous renvoyons le lecteur, par exemple, aux références [30, 32, 36, 56].

Remarque 1.3.9. Deux points faibles de l'algorithme du type gradient sont :

- (a) L'algorithme peut nécessiter de nombreuses itérations pour converger vers le minimum.
- (b) La recherche du pas optimal, généralement effectuée par une recherche linéaire, peut se révéler très longue.

Les méthodes simpliciales

1. Méthode du gradient réduit (Wolfe)

Parmi les méthodes simpliciales on cite, celle de gradient réduit qui est proposée par Wolfe en 1962. C'est une extension directe de la méthode du simplexe [44], appliquée à la résolution d'un programme quadratique de la forme **(FS)**. Dans ses grandes lignes, la méthode est la suivante :

Algorithme 3 : Algorithme du gradient réduit.

Initialisation :

Soit x^0 est un point réalisable ;

$k = 0$;

Etape 1 :

I_k l'ensemble des indices des composantes positives de x^k .

$B = A_j : j \in I_k$,

$N = A_j : j \notin I_k$,

$r_N^k = \nabla_N f(x) - N^* \nabla_B f(x)$, avec $N^* = B^{-1}N$.

$$d_N^k = \begin{cases} -r_j^k & \text{si } r_j^k \leq 0, \\ -x_j^k r_j^k & \text{si } r_j^k > 0. \end{cases}$$

Si $d_N^k = 0$ **alors**

x^k est un point de Kuhn-Tucker (stop).

Sinon

calculer $d_B^k = -B^{-1}N d_N^k$.

Fin

Etape 2 :

Résoudre le sous problème unidimensionnel : $\alpha^* = \min_{0 \leq \alpha \leq \alpha_{\max}} \phi(\alpha) = f(x^k + \alpha d^k)$,

où

$$\alpha_{\max} = \begin{cases} \min_{j=1, \dots, n} \left\{ \frac{-x_j^k}{d_j^k}, \text{ si } d_j^k < 0 \right\}, \\ +\infty \text{ si } d_j^k \geq 0. \end{cases}$$

$x^{k+1} = x^k + \alpha^* d^k$;

$k \leftarrow k + 1$; et on revient à l'étape 1.

De plus, nous avons le résultat de convergence suivant :

Théorème 1.3.10. ([12]) Si $f \in C^1(D)$ et sous les hypothèses de non dégénérescence alors la suite $(x^k)_{k \in \mathbb{N}}$ de Wolfe converge vers un point de Karush-Kuhn-Tucker.

Pour plus de détails sur les résultats résumés pour cette méthode, nous renvoyons par exemple aux références [12, 56].

Remarque 1.3.11. *Les inconvénients des méthodes simpliciales sont le recyclage et la complexité exponentielle. Plusieurs chercheurs ont suggéré des méthodes de surmonter ces inconvénients, parmi lesquelles les méthodes de points intérieurs que nous aborderons dans le paragraphe suivant.*

Les méthodes de points intérieurs

L'étude des méthodes de points intérieurs (*MPIs*) est actuellement l'une des plus actifs domaines de recherche en optimisation. Le nom méthodes de points intérieurs provient du fait que les points générés par cette méthode se trouvent à l'intérieur du domaine réalisable. Cela contraste avec la méthode du simplexe où les itérés se déplacent le long de la limite de la région réalisable d'un point extrême à un autre. De nos jours, les *MPIs* ont devenues assez riche en théorie et ont été appliqués à des problèmes pratiques avec un succès extraordinaire. Nous présentons dans la suite les quatre classes fondamentales des méthodes de points intérieurs.

1. Méthodes affines

En 1967, Dikin [27] est le premier qui introduit ce type des méthodes de points intérieurs pour (*PQC*). Après l'apparition de l'article de Karmarkar [42] en 1984 pour l'optimisation linéaire (*OL*) plusieurs chercheurs sont consacré leurs travaux aux méthodes affines parmi eux : Barnes [10], Cavalier et Soyster [21], Vanderbei et Freedman [76]. La convergence de ces méthodes a été étudiée premièrement par Dikin qui a prouvé que l'algorithme convergent sous l'hypothèse de non dégénérescence. Puis, Tseng et Luo [75] ont montré que la méthode converge même dans le cas dégénéré. Il y a trois types des méthodes affines : primal, dual et primal-dual.

Pour des détails sur les méthodes affines, on peut renvoyer le lecteur par exemple à [40, 27].

2. Méthodes projectives

L'algorithme du simplexe est restée longtemps l'algorithme de référence pour la programmation linéaire malgré sa complexité exponentielle jusqu'à l'apparition du célèbre article de Karmarkar [42] en 1984 qui contient un nouvel

algorithme polynomial. L'algorithme des méthodes projectives a été proposé à l'origine par Narendra Karmarkar. Cet algorithme a été essentiellement différent de la méthode du simplexe par le fait que, dans cette dernière, on se déplace en suivant la frontière du domaine réalisable, alors que pour la méthode de Karmarkar, on progresse tout en restant strictement à l'intérieur du domaine réalisable.

L'algorithme de Karmarkar avait un grand intérêt théorique puisqu'il est polynomial ; en effet, il exige un $O(nL)$ itérations et chaque itération exige $O(n^3)$ opérations arithmétiques, d'où une complexité totale de $O(n^4L)$ opérations. Karmarkar a même été capable de la réduire à $O(n^{3.5}L)$ opérations.

Nombreux chercheurs ont suggéré des modifications de l'algorithme original de Karmarkar afin d'aboutir à un algorithme plus efficace et plus utile dans la pratique parmi eux Todd et Burrell [73], Ye et Kojima [81] et Bouafia [13].

En 1989, Yinyu et al. [82] ont fait une extension de la méthode de Karmarkar pour résoudre un problème de la programmation quadratique convexe et ceci pour leur grand succès en pratique.

Pour plus de détails sur les méthodes projectives, nous renvoyons le lecteur, par exemple, aux références [40, 44, 55].

3. Méthodes de réduction du potentiel

Une autre variante de l'algorithme de Karmarkar est la méthode potentielle. Les algorithmes se distinguent par le fait qu'ils gardent les caractéristiques de l'algorithme projectif : notamment, la borne de la complexité polynomiale et l'habilité à prendre de grandes étapes basées sur la fonction potentielle sans l'utilisation du changement de variables projectif. Le premier algorithme de ce type a été proposé par Gonzaga [38]. Puis, Ye [79] a proposé un algorithme analogue qui utilise une fonction potentielle primale-duale pour obtenir un algorithme de complexité meilleure que celle mentionnée précédemment. Les chercheurs ont proposé des fonctions potentielles de type primal, dual et primal-dual, parmi lesquelles :

-La première fonction potentielle primale en programmation linéaire est in-

troduite par Karmarkar [42] est définie par :

$$f(x, Z) = \rho \log(c^t x - Z) - \sum_{i=1}^n \log x_i,$$

où Z est une borne inférieure de la valeur optimale de la fonction objectif et $\rho = n + 1$.

-La fonction potentielle duale est introduite par Gonzaga [38] est définie par :

$$f(y, Z) = \rho \log(Z - b^t y) - \sum_{i=1}^n \log z_i,$$

où Z est une borne supérieure de la valeur optimale de la fonction objectif et $\rho = n + 1$.

-Todd et Ye [74] ont été les premiers à introduire une fonction potentielle primale-duale définie comme suit :

$$f(x, z) = \rho \log(x^t z) - \sum_{i=1}^n \log x_i - \sum_{i=1}^n \log z_i,$$

où $\rho = n + \sqrt{n}$.

Pour avoir plus de détails sur la méthode figurant dans ce paragraphe, le lecteur pourra consulter par exemple [40].

4. Méthodes de trajectoire centrale

Comme son nom l'indique, cette méthode est basée sur le suivi d'une trajectoire centrale. La méthode de trajectoire centrale consiste à rester dans un certain voisinage de la trajectoire centrale en utilisant des itérations de Newton. Il existe deux approches différentes de ces méthodes : une approche primale et une autre primale-duale. Pour mettre en œuvre chacune de ces approches, nous pouvons utiliser soit des algorithmes à courts-pas, soit des algorithmes à long-pas. Tout dépend de la manière de mise à jour des paramètres. Les méthodes primales-duales sont les plus utilisées en pratique à cause de leur efficacité numérique.

La méthode de trajectoire centrale a été étudiée en premier pour la programmation linéaire par Bayer et Lagarias [11] ensuite par Meggido [54], Gonzaga

[38], Monteiro et Adler [57], Kojima et al. [48], Roos et Via1 [71].

A cause de leurs propriétés attractives, notamment la complexité polynomiale et la convergence super linéaire, nombreux chercheurs ont proposé des extensions intéressantes pour résoudre les problèmes de la programmation quadratique convexe. Achache et Goutali [3] ont proposé une méthode de trajectoire centrale réalisable primale-duale à court pas à itération de Newton complet. Ils ont proposé deux nouveaux paramètres qui décrivent le paramètre barrière et le seuil qui mesure la taille du voisinage de la trajectoire centrale. Mais, il n'existe aucune technique en pratique pour obtenir un point initiale strictement réalisable voisin de la trajectoire centrale pour démarrer leur algorithme. Pour remédier à cette difficulté, une méthode de trajectoire centrale de type non réalisable a été proposée par Roumili et al. [68, 69]. Kebbiche et Djefal [26, 43] ont introduit le paramètre poids, associé à la fonction barrière pour forcer le point initial dans la trajectoire centrale. L'inconvénient de cette dernière alternative est le choix du paramètre de mise à jour θ , jusqu'à présent, nous n'avons aucun résultat confirmé que sa valeur obtenue par les chercheurs est la meilleure.

Dans cette thèse, nous nous intéressons aux méthodes de points intérieurs de type trajectoire centrale. Maintenant, nous allons présenter un aperçu sur ces méthodes appliquées à la résolution d'un problème quadratique convexe.

Méthode de trajectoire centrale classique

Reprenons le problème primal de l'optimisation quadratique convexe au format standard :

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Q x + c^t x, \\ Ax = b, \\ x \geq 0. \end{cases} \quad (PQC)$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique semi-définie positive, $A \in \mathbb{R}^{m \times n}$ tel que $\text{rang}(A) = m < n$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

Et son dual donné par :

$$\begin{cases} \max -\frac{1}{2}x^t Qx + b^t y, \\ A^t y + z - Qx = c, \\ z \geq 0, x \geq 0. \end{cases} \quad (DQC)$$

où $y \in \mathbb{R}^m, z \in \mathbb{R}^n$.

C'est bien connu, que trouver une solution optimale de (PQC) et (DQC) est équivalent à résoudre le système d'équations non linéaire suivant :

$$\begin{cases} A^t y + z - Qx = c, z \geq 0, \\ Ax = b, x \geq 0, \\ xz = 0. \end{cases} \quad (KKT)$$

Ces conditions sont appelées les conditions de Karush-Kuhn-Tucker (KKT). La première et la deuxième condition représentent la condition de faisabilité duale et primale, respectivement. Tandis que la troisième équation est la condition de complémentarité pour (PQC) et (DQC).

On associe à (PQC) le problème perturbé (PQC_μ) suivant :

$$\begin{cases} \min f_{\mu}(x) = \frac{1}{2}x^t Qx + c^t x - \mu \sum_{i=1}^n \ln x_i, \\ Ax = b, \\ x > 0, \end{cases} \quad (PQC_{\mu})$$

où μ est le paramètre barrière tel que $\mu > 0$ et $f_{\mu}(x)$ est la fonction barrière (pénalisée).

◆ Si $\mu \rightarrow 0$ alors la solution de (PQC_μ) converge vers la solution de (PQC), donc on diminue les valeurs de μ et on résout la suite des problèmes (PQC_μ) jusqu'à l'obtention d'une solution optimale de (PQC).

Lemme 1.3.12. f_{μ} est une fonction strictement convexe.

Démonstration. Reprenons f_{μ} comme suit :

$$f_{\mu}(x) = \frac{1}{2}x^t Qx + c^t x - \mu \sum_{i=1}^n \ln x_i, \mu > 0,$$

il est clair que $f_\mu(x) \in C^\infty(\mathfrak{R}_{++}^n)$, donc on a :

$$\begin{aligned}\nabla f_\mu(x) &= Qx + c - \mu X^{-1}e, \text{ avec } e = (1, \dots, 1)^t \in \mathfrak{R}^n, X = \text{diag}(x_1, \dots, x_n), \\ \nabla^2 f_\mu(x) &= Q + \mu X^{-2},\end{aligned}$$

où Q est une matrice symétrique semi définie positive, X est une matrice diagonale définie positive car $x > 0$ et $\mu > 0$ alors $\nabla^2 f_\mu(x)$ est une matrice définie positive, donc f_μ est strictement convexe. Ce qui achève la preuve du lemme. ■

Lemme 1.3.13. ([58]) *Sous l'hypothèse **(H1)** et $f_\mu(x)$ est une fonction strictement convexe, le problème (PQC_μ) admet une solution optimale unique pour tout $\mu > 0$.*

◆ Pour le problème (PQC_μ) les conditions de *KKT* sont à la fois nécessaires et suffisantes et s'écrivent comme suit :

$$\begin{cases} Qx + c - \mu X^{-1}e - A^t y = 0, \\ Ax = b, x > 0 \end{cases}$$

où $y \in \mathfrak{R}^m$ est le multiplicateur de Lagrange associé à la contrainte $Ax = b$ du problème (PQC_μ) .

Mettons $z = \mu X^{-1}e \in \mathfrak{R}_{++}^n$, le système précédent devient :

$$\begin{cases} A^t y + z - Qx = c, z > 0, \\ Ax = b, x > 0, \\ Xz - \mu e = 0. \end{cases} \quad (KKT_\mu)$$

où $X = \text{diag}(x_1, \dots, x_n)$.

◆ Le système (KKT_μ) représente aussi les conditions d'optimalité du problème dual paramétré suivant :

$$\begin{cases} \max -\frac{1}{2}x^t Qx + b^t y + \mu \sum_{i=1}^n \ln x_i, \\ A^t y + z - Qx = c, \\ z > 0, x > 0. \end{cases} \quad (DQC_\mu)$$

Définition 1.3.14. — Soit $(x(\mu), y(\mu), z(\mu))$ la solution du système (KKT_μ) , la trajectoire centrale de (PQC_μ) est définie par l'ensemble suivant :

$$TC = \{(x(\mu), y(\mu), z(\mu)), \mu > 0\}.$$

— Pour mesurer la qualité de la solution trouvée $(x(\mu), y(\mu), z(\mu))$, on introduit un facteur de centralité ou de proximité qui est défini par la formule suivante :

$$\delta(x, z; \mu) = \|Xz - \mu e\|. \quad (1.3.2)$$

— On dit que la solution $(x(\mu), y(\mu), z(\mu))$ est voisine de la trajectoire centrale si elle appartient à l'ensemble

$$V(\theta) = \{(x(\mu), y(\mu), z(\mu)) \in F^0 : \delta(x, z; \mu) \leq \theta\mu, 0 < \theta < 1\}.$$

Puisque le système (KKT_μ) est non linéaire, il est difficile d'obtenir une solution exacte de (PQC_μ) . Nous cherchons une solution approximative par la méthode de Newton qui est la meilleure méthode pour la résolution à cause de sa vitesse de convergence quadratique. Donc, on applique la méthode de Newton sur l'équation $G_\mu(x, y, z) = 0$ pour chaque $\mu > 0$, où

$$G : \mathfrak{R}^{2n+m} \rightarrow \mathfrak{R}^{2n+m}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} A^t y + z - Qx - c \\ Ax - b \\ Xz - \sigma \mu e \end{pmatrix},$$

tel que $0 < \sigma < 1$ et $(x, y, z) \in F^0$.

Le nouveau itéré par la méthode de Newton s'écrit sous la forme :

$$\bar{x} = x + \Delta x,$$

$$\bar{y} = y + \Delta y,$$

$$\bar{z} = z + \Delta z,$$

où $(\Delta x, \Delta y, \Delta z)$ est la direction de Newton qui est la solution du système linéaire

$$\nabla G_\mu(x, y, z) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -G_\mu(x, y, z),$$

tel que :

$$\nabla G_\mu(x, y, z) = \begin{pmatrix} -Q & A^t & I \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix}.$$

où $Z = \text{diag}(z_1, \dots, z_n)$.

Par un simple calcul, le système précédent peut s'écrire sous la forme suivante :

$$\begin{cases} A^t \Delta y + \Delta z - Q \Delta x = 0, \\ A \Delta x = 0, \\ Z \Delta X + X \Delta z = \sigma \mu e - X z. \end{cases} \quad (\text{SN})$$

On réitère jusqu'à l'obtention d'une valeur proche de zéro de μ .

L'algorithme de trajectoire centrale est donc donné comme suit :

Algorithme 4 : Méthode de la trajectoire centrale réalisable classique.

Initialisation :

Un paramètre de précision $\varepsilon > 0$.

Un paramètre de mise à jour $\theta, 0 < \theta < 1$.

Un paramètre de seuil $\beta, 0 < \beta < 1$.

Un point initial strictement réalisable (x^0, y^0, z^0) et $\mu^0 = \frac{(x^0)^t z^0}{n}$ tel que :

$$\delta(x^0, z^0; \mu^0) \leq \beta.$$

$$k = 0.$$

Tant que $n\mu > \varepsilon$ **do**

(a) Résoudre le système (SN) pour trouver la direction de Newton $(\Delta x^k, \Delta y^k, \Delta z^k)$.

$$(b) (x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + (\Delta x^k, \Delta y^k, \Delta z^k);$$

$$(c) \mu^{k+1} = (1 - \theta) \mu^k;$$

$$k \leftarrow k + 1;$$

Fin Tant que

Pour plus de détails sur cette méthode, nous renvoyons par exemple au référence [3].

Conclusion 1.3.15. *Ces progrès rivalisant, n'échappent pas tout de même aux inconvénients théoriques, algorithmiques et numériques, entre autre, le problème d'initialisation qui devient de plus en plus difficile en passant aux autres problèmes mathématiques (la question qui se pose comment choisir un point initial strictement réalisable près de la trajectoire centrale pour démarrer l'algorithme?). Ce problème d'initialisation est au centre de notre étude, il figure depuis des années au menu de la recherche dans le domaine de la programmation mathématique. Pour atténuer à cet inconvénient, une alternative particulièrement intéressante pour nous est celle des MPIs basées sur les fonctions barrières qui sont définies par une grande classe de fonctions uni-variées appelées fonctions noyaux qui ont récemment été utilisés avec succès pour concevoir de nouveaux. Nous étudions dans le chapitre suivant, une méthode de trajectoire centrale primale-duale basée sur les fonctions noyaux. Cette approche a l'avantage de démarrer avec un point de départ (x^0, y^0, z^0) strictement réalisable non nécessairement près de la trajectoire centrale.*

CHAPITRE 2

SURVOL SUR LES FONCTIONS NOYAUX ADAPTÉES AUX MÉTHODES DE POINTS INTÉRIEURS PRIMALES-DUALES DE TYPE TRAJECTOIRE CENTRALE

Aperçu historique

Les fonctions noyaux jouent un rôle important dans la conception et l'analyse des méthodes des points intérieurs (*MPIs*). Elles ne sont pas seulement utilisées pour déterminer les directions de recherche mais aussi pour mesurer la distance entre l'itération donnée et le μ -centre.

Actuellement, les *MPIs* basées sur les fonctions noyaux sont parmi les méthodes efficaces pour la résolution des problèmes d'(OL) qui ont été étudié par plusieurs chercheurs dans [2, 7, 14, 15, 17, 22, 25, 29, 46, 51, 60, 59, 61, 63]. Les *MPIs* basées sur les fonctions noyaux sont également étendus à nombreux domaines d'op-

timisation mathématique tels que l'optimisation quadratique semi-définie convexe [39], l'optimisation de complémentarité linéaire semi-définie monotone [5] et l'optimisation semi-définie [28].

Encouragés par la stratégie envisagée dans [7] qui est donnée par Bai et al. pour l'OL, nous suivrons les étapes utilisées pour établir l'analyse de la MPI basée sur une fonction noyau pour la résolution d'un programme quadratique convexe. Donc, l'objectif de ce chapitre est de faire une extension d'une méthode de trajectoire centrale basée sur une fonction noyau pour la programmation linéaire à la programmation quadratique convexe. Nous donnons en premier, un rappel sur les notions de base des fonctions noyaux. Ensuite, nous présentons dans un contexte général le principe, l'algorithme et l'analyse de la complexité de l'algorithme obtenu de trajectoire centrale moyennant les fonctions noyaux.

2.1 Fonctions noyaux et leurs propriétés

Nous donnons dans cette section la définition d'une fonction noyau et quelques propriétés concernant les conditions de qualification.

Nous appelons $\psi :]0, +\infty[\rightarrow [0, +\infty[$ une fonction noyau si elle est deux fois différentiable et vérifie les conditions suivantes :

- (a) $\psi(1) = \psi'(1) = 0,$
- (b) $\psi''(t) > 0, \forall t > 0,$
- (c) $\lim_{t \downarrow 0} \psi(t) = \lim_{t \rightarrow +\infty} \psi(t) = +\infty.$

Notons que les conditions **(a)** et **(b)** signifient que ψ est une fonction strictement convexe et admet un minimum en $t = 1$. La condition **(c)** crée une barrière aux bords du domaine de définition, donc ψ porte le nom de fonction barrière. On peut écrire ψ par sa dérivée seconde comme suit :

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\zeta) d\zeta d\xi.$$

Nous pouvons écrire aussi $\psi(t)$ comme suit :

$$\psi(t) = \psi_c(t) + \psi_b(t),$$

où $\psi_c(t)$ est le terme de croissance et $\psi_b(t)$ est le terme barrière de fonction noyau. Le terme de croissance caractérise le comportement de $\psi(t)$ lorsque t tend vers $+\infty$, tandis que le terme barrière caractérise son comportement lorsque t s'approche de zéro. Dans tous les cas, le terme barrière est décroissant de façon monotone en t .

2.1.1 Conditions de qualification d'une fonction noyau

Définition 2.1.1. Soit $\psi \in C^3(\mathbb{R}_{++})$ une fonction noyau, on dit que ψ est qualifiée (éligible) si elle satisfait les conditions suivantes :

(C.g.a) $t\psi''(t) + \psi'(t) > 0, \forall t < 1,$

(C.g.b) $t\psi''(t) - \psi'(t) > 0, \forall t > 1,$

(C.g.c) $\psi'''(t) < 0, \forall t > 0,$

(C.g.d) $2(\psi''(t))^2 - \psi'(t)\psi'''(t) > 0, \forall t < 1,$

(C.g.e) $\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \forall \beta > 1, \forall t > 1.$

Nous signalons que les quatre premières conditions sont logiquement indépendantes et que la cinquième condition est une conséquence de **(C.g.b)** et **(C.g.c)** qui est l'objective du Lemme 2.1.7 ci-dessous. Par conséquence, il est facile de savoir que $\psi(t)$ est éligible si elle satisfait les quatre premières conditions.

Les deux lemmes suivants indiquent clairement que les conditions **(C.g.a)** et **(C.g.b)** admettent une belle interprétation qui est la propriété de convexité exponentielle, ou plus brièvement e-convexité. Cette propriété est importante et essentielle dans l'analyse des algorithmes des points intérieurs basés sur les fonctions noyaux.

Lemme 2.1.2. ([28]) Soit $\psi \in C^2(\mathbb{R}_{++})$, alors les propriétés suivantes sont équivalentes :

(i) La fonction ϕ définie par $\phi(\xi) = \psi(e^\xi)$ est convexe, $\forall \xi \in \mathbb{R}$.

(ii) $\psi(\sqrt{t_1 t_2}) \leq \frac{\psi(t_1) + \psi(t_2)}{2}, \forall t_1, t_2 > 0.$

(iii) $t\psi''(t) + \psi'(t) \geq 0, \forall t > 0.$

Démonstration. Montrons que (i) \iff (ii) : nous savons que la fonction $\phi(\xi) = \psi(e^\xi)$ est une fonction continue, donc pour être convexe, il suffit d'être mid-convexe.

$$\psi\left(e^{\frac{\xi_1 + \xi_2}{2}}\right) \leq \frac{1}{2} \left(\psi(e^{\xi_1}) + \psi(e^{\xi_2}) \right), \forall \xi_1, \xi_2 \in \mathfrak{R}.$$

Mettons $t_1 = e^{\xi_1}$ et $t_2 = e^{\xi_2}$, Il est évident qu'on a $t_1, t_2 > 0$, et la relation ci-dessus peut s'écrire comme suit

$$\psi(\sqrt{t_1 t_2}) \leq \frac{\psi(t_1) + \psi(t_2)}{2}, \forall t_1, t_2 > 0.$$

Concernant l'équivalence (i) \iff (iii) :

$$\begin{aligned} \phi \text{ convexe} &\iff \phi''(\xi) = \left(\psi(e^\xi)\right)'' = e^\xi \left(e^\xi \psi''(e^\xi) + \psi'(e^\xi) \right) \geq 0, \forall \xi \in \mathfrak{R}, \\ &\iff t \left(t\psi''(t) + \psi'(t) \right) \geq 0, \forall t = e^\xi > 0, \\ &\iff t\psi''(t) + \psi'(t) \geq 0, \forall t > 0. \end{aligned}$$

Ce qui conclut la preuve du lemme. ■

Lemme 2.1.3. ([28]) Soit $\psi \in C^2(\mathfrak{R}_{++})$, alors les propriétés suivantes sont équivalentes :

(i) La fonction ϕ définie par $\phi(\xi) = \psi(\sqrt{\xi})$ est convexe, $\forall \xi > 0.$

(ii) $\psi\left(\sqrt{\frac{t_1^2 + t_2^2}{2}}\right) \leq \frac{\psi(t_1) + \psi(t_2)}{2}, \forall t_1, t_2 > 0.$

(iii) $t\psi''(t) - \psi'(t) > 0, \forall t > 0.$

Démonstration. Montrons que (i) \iff (ii) : nous savons que la fonction $\phi(\xi) = \psi(\sqrt{\xi})$ est une fonction continue, donc pour être convexe, il suffit d'être mid-convexe. Donc, nous avons :

$$\psi\left(\sqrt{\frac{\xi_1 + \xi_2}{2}}\right) \leq \frac{1}{2} \left(\psi(\sqrt{\xi_1}) + \psi(\sqrt{\xi_2}) \right), \forall \xi_1, \xi_2 \in \mathfrak{R}.$$

Mettons $t_1 = \sqrt{\xi_1}$ et $t_2 = \sqrt{\xi_2}$, il est évident qu'on a $t_1, t_2 > 0$, et la relation ci-dessus peut s'écrire comme suit

$$\psi \left(\sqrt{\frac{t_1^2 + t_2^2}{2}} \right) \leq \frac{\psi(t_1) + \psi(t_2)}{2}, \forall t_1, t_2 > 0.$$

Concernant l'équivalence (i) \iff (iii) :

$$\begin{aligned} \phi \text{ convexe} &\iff \phi''(\xi) = \frac{1}{4\xi^{\frac{3}{2}}} \left(\sqrt{\xi} \psi''(\sqrt{\xi}) - \psi'(\sqrt{\xi}) \right) \geq 0, \forall \xi \in \mathfrak{R}, \\ &\iff \frac{1}{4t^3} \left(t \psi''(t) - \psi'(t) \right) \geq 0, \forall t = \sqrt{\xi} > 0, \\ &\iff t \psi''(t) + \psi'(t) \geq 0, \forall t > 0. \end{aligned}$$

Ce qui termine la preuve du lemme. ■

Lemme 2.1.4. ([28]) Soit ψ une fonction noyau qui satisfait la condition **(C.9.c)**, alors pour tout $t < 1$ on a :

$$\begin{aligned} \psi'(t) < \psi''(t)(t-1) \text{ et } \psi(t) < \frac{1}{2} \psi'(t)(t-1), \\ \frac{1}{2} \psi''(1)(t-1)^2 < \psi(t) < \frac{1}{2} \psi''(t)(t-1)^2. \end{aligned}$$

Démonstration. Nous considérons pour tout $t < 1$ la fonction $g(t)$ qui est définie par :

$$g(t) = \psi'(t)(t-1) - 2\psi(t),$$

il est clair que $g(1) = 0$ et

$$g'(t) = \psi''(t)(t-1) - \psi'(t),$$

on a aussi $g'(1) = 0$ et

$$g''(t) = \psi'''(t)(t-1).$$

En utilisant la condition **(C.9.c)** nous obtenons :

$$g'(t) > 0 \text{ et } g(t) > 0,$$

d'où :

$$\psi'(t) < \psi''(t)(t-1) \text{ et } \psi(t) < \frac{1}{2}\psi'(t)(t-1).$$

En utilisant le développement de Taylor au voisinage de $t = 1$ et $\psi(1) = \psi'(1) = 0$, nous trouvons :

$$\begin{aligned} \psi(t) &= \psi(1) + \psi'(1)(t-1) + \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{6}\psi'''(\zeta)(t-1)^3, \quad t < \zeta < 1, \\ &= \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{6}\psi'''(\zeta)(t-1)^3, \end{aligned}$$

et

$$\psi(t) = \frac{1}{2}\psi''(\zeta)(t-1)^2, \quad t < \zeta < 1.$$

Comme $\psi'''(\zeta) < 0$ et $t < 1$, nous obtenons :

$$\frac{1}{2}\psi''(1)(t-1)^2 < \psi(t) < \frac{1}{2}\psi''(t)(t-1)^2.$$

Ce qui conclut la preuve du lemme. ■

Lemme 2.1.5. ([28]) Soit ψ une fonction noyau qui satisfait la condition **(C.9.c)**, alors pour tout $t > 1$ on a :

$$\psi'(t) > \psi''(t)(t-1) \text{ et } \psi(t) > \frac{1}{2}\psi'(t)(t-1),$$

$$\frac{1}{2}\psi''(t)(t-1)^2 < \psi(t) < \frac{1}{2}\psi''(1)(t-1)^2.$$

Démonstration. Nous définissons la fonction $h(t)$ comme suit :

$$h(t) = 2\psi(t) - \psi'(t)(t-1), \quad \forall t > 1,$$

il est clair que $h(1) = 0$ et

$$h'(t) = \psi'(t) - \psi''(t)(t-1),$$

on a $h'(1) = 0$ et

$$h''(t) = -\psi'''(t)(t-1).$$

En utilisant la condition **(C.g.c)** nous trouvons :

$$h'(t) > 0 \text{ et } h(t) > 0,$$

donc

$$\psi'(t) > \psi''(t)(t-1) \text{ et } \psi(t) > \frac{1}{2}\psi'(t)(t-1).$$

En utilisant le développement de Taylor au voisinage de $t = 1$ et puisque $\psi(1) = \psi'(1) = 0$, nous avons :

$$\begin{aligned} \psi(t) &= \psi(1) + \psi'(1)(t-1) + \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{6}\psi'''(\zeta)(t-1)^3, \quad 1 < \zeta < t, \\ &= \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{6}\psi'''(\zeta)(t-1)^3, \end{aligned}$$

et

$$\psi(t) = \frac{1}{2}\psi''(\zeta)(t-1)^2, \quad 1 < \zeta < t.$$

Comme $\psi'''(\zeta) < 0$ et $t > 1$, nous obtenons :

$$\frac{1}{2}\psi''(t)(t-1)^2 < \psi(t) < \frac{1}{2}\psi''(1)(t-1)^2.$$

Ce qui conclut la preuve du lemme. ■

2.1.2 Propriétés des fonctions noyaux

Lemme 2.1.6. ([28]) Soit ψ une fonction noyau alors on a :

- (a) $t\psi'(t) \geq \psi(t), \forall t \geq 1.$
- (b) $\psi'(t_1) \leq 0, \psi'(t_2) \geq 0,$ avec $0 < t_1 \leq 1 \leq t_2.$

Démonstration. Pour montrer (a), on considère :

$$h(t) = t\psi'(t) - \psi(t), \forall t \geq 1.$$

On a $h(1) = 0$ et $h'(t) = t\psi''(t) \geq 0$ donc $h(t) \geq 0, \forall t \geq 1.$

Pour montrer (b), en utilisant (a), on a

$$t_2 \psi'(t_2) \geq \psi(t_2), 1 \leq t_2.$$

Maintenant en utilisant le Lemme 2.1.4, on obtient :

$$\psi'(t_1) < \psi''(t_1)(t_1 - 1), t_1 \leq 1.$$

Comme $\psi''(t) > 0, \forall t > 0$, on trouve :

$$\psi'(t_1) \leq 0, \psi'(t_2) \geq 0.$$

Ce qui termine la preuve du Lemme 2.1.6. ■

Lemme 2.1.7. ([28]) Si $\psi(t)$ satisfait les conditions **(Q.C.b)** et **(Q.C.c)**, alors $\psi(t)$ satisfait la condition **(Q.C.e)**.

Démonstration. Pour $t > 1$, nous considérons :

$$f(\beta) = \psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t), \forall \beta > 1.$$

Notons que $f(1) = 0$. De plus, on a :

$$f'(\beta) = \psi'''(\beta t)(t\psi''(t) - \psi'(t)) - \beta t\psi'(t)\psi'''(\beta t), \forall \beta > 1,$$

puisque ψ vérifie les conditions **(Q.C.b)** et **(Q.C.c)** et en utilisant (b) du lemme 2.1.6 pour $t > 1$, nous déduisons que $f(\beta)$ est une fonction croissante. Par conséquent, $f(\beta) > 0$ pour $\beta > 1$. Ce qui termine la preuve du lemme. ■

Lemme 2.1.8. ([28]) Supposons que $\psi(t_1) = \psi(t_2)$ avec $t_1 \leq 1 \leq t_2$. Alors on a :

- (1) $\psi(\beta t_1) \leq \psi(\beta t_2), \forall \beta \geq 1$,
- (2) $\psi(\beta t_1) = \psi(\beta t_2)$ si $t_1 = t_2 = 1$ ou $\beta = 1$.

Démonstration. Nous supposons que $\psi(t_1) = \psi(t_2)$ avec $t_1 \leq 1 \leq t_2$:

Pour montrer (1), on définit $f(\beta)$ par :

$$f(\beta) = \psi(\beta t_1) - \psi(\beta t_2), \quad t_1 \leq 1 \leq t_2, \quad \forall \beta \geq 1.$$

On a $f(1) = 0$ et

$$f'(\beta) = t_1 \psi'(\beta t_1) - t_2 \psi'(\beta t_2).$$

Puisque $\psi''(t) > 0, \forall t > 0$, alors $\psi'(t)$ est strictement croissante. Pour cela $\psi'(\beta t_1) \leq \psi'(\beta t_2)$. Donc on a :

$$f'(\beta) = t_1 \psi'(\beta t_1) - t_2 \psi'(\beta t_2) \leq t_1 \psi'(\beta t_2) - t_2 \psi'(\beta t_2) = (t_1 - t_2) \psi'(\beta t_2),$$

comme $t_1 \leq t_2$ et $\psi'(t) \geq 0, \forall t \geq 1$, donc $f'(\beta) \leq 0, \forall \beta \geq 1$. Alors :

$$f(\beta) = \psi(\beta t_1) - \psi(\beta t_2) \leq 0.$$

Pour montrer (2), d'une part si $\beta = 1$ alors par évidence l'égalité est satisfaite. D'autre part, si $\beta > 1$ et $f(\beta) = 0$ alors par le théorème de la valeur moyenne implique que

$$f'(\xi) = 0, \quad \text{pour } \xi \in (1, \beta),$$

cela implique

$$\psi'(\xi t_1) = \psi'(\xi t_2).$$

Par le Lemme 2.1.6, la fonction $\psi'(t)$ est strictement monotone, cela implique que $\xi t_1 = \xi t_2$, d'où $t_1 = t_2$. Puisque $t_1 \leq 1 \leq t_2$, on obtient $t_1 = t_2 = 1$.

Ce qui conclut la preuve du Lemme 2.1.8. ■

2.2 Présentation de la méthode

Dans cette thèse, nous considérons le problème primal de l'optimisation quadratique convexe au format standard :

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Q x + c^t x, \\ Ax = b, \\ x \geq 0, \end{cases} \quad (PQC)$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique semi-définie positive, $A \in \mathbb{R}^{m \times n}$ tel que $\text{rang}(A) = m < n$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$. Et son dual :

$$\begin{cases} \max -\frac{1}{2}x^t Q x + b^t y, \\ A^t y + z - Q x = c, \\ z \geq 0, x \geq 0. \end{cases} \quad (DQC)$$

où $y \in \mathbb{R}^m, z \in \mathbb{R}^n$.

Il est bien connu que la recherche d'une solution optimale de (PQC) et (DQC) est équivalente à la résolution des conditions d'optimalité de Karush-Kuhn-Tucker suivantes :

$$\begin{cases} A^t y + z - Q x = c, z > 0, \\ Ax = b, x > 0, \\ Xz = 0. \end{cases} \quad (2.2.1)$$

L'idée de base qui sous-tend les MPIs primales-duales est de remplacer la condition de complémentarité $Xz = 0$ dans (2.2.1) par l'équation perturbée $Xz = \mu e, \mu > 0$. Ainsi, nous obtenons le système perturbé suivant :

$$\begin{cases} A^t y + z - Q x = c, z > 0, \\ Ax = b, x > 0, \\ Xz - \mu e = 0. \end{cases} \quad (2.2.2)$$

Nous supposons que (PQC) et (DQC) satisfont la condition de points intérieurs (CPI), c'est-à-dire, il existe (x^0, y^0, z^0) tel que :

$$\begin{cases} Ax^0 = b, x^0 > 0, \\ A^t y^0 + z^0 - Q x^0 = c, z^0 > 0, \end{cases} \quad (2.2.3)$$

alors le système paramétré (2.2.2) a une solution unique $(x(\mu), y(\mu), z(\mu))$ pour chaque $\mu > 0$. $x(\mu)$ est appelé le μ -centre de (PQC) et $(y(\mu), z(\mu))$ est le μ -centre de (DQC). L'ensemble de tous les μ -centres forme ce qu'on appelle la trajectoire centrale de (PQC) et (DQC).

Si $\mu \rightarrow 0$, alors la limite du chemin central existe et puisque le point limite

satisfait la condition de complémentarité, la limite donne une solution optimale de (PQC) et (DQC). Pour plus détails sur ce qui se précède voir [58].

Sans perte de généralité, nous supposons que le point $(x(\mu), y(\mu), z(\mu))$ est connu pour $\mu > 0$. Par exemple, on peut supposer que $\mu^0 = 1$. Nous réduisons ensuite $\mu = (1 - \theta)\mu, 0 < \theta < 1$. Nous appliquons la méthode de Newton pour résoudre itérativement les équations non linéaires (2.2.2). Donc, nous avons le système de Newton suivant :

$$\begin{cases} A^t \Delta y + \Delta z - Q \Delta x = 0, \\ A \Delta x = 0, \\ Z \Delta X + X \Delta z = X z - \mu e, \end{cases} \quad (2.2.4)$$

comme A est de rang plein, le système (2.2.4) admet une solution unique $(\Delta x, \Delta y, \Delta z)$ dite la direction de Newton, qui est utilisée dans toutes les implémentations existantes des *MPIs* primales-duales.

Le nouveau itéré de Newton avec le pas α fixé ($0 < \alpha \leq 1$) est donné par :

$$x^+ = x + \alpha \Delta x, \quad y^+ = y + \alpha \Delta y, \quad z^+ = z + \alpha \Delta z. \quad (2.2.5)$$

Le choix du pas α est un autre point crucial de l'algorithme. Il faut faire en sorte que la proximité des itérés avec le μ -centre s'améliore suffisamment.

Dans cette thèse, nous suivons [7] et reformulons cette approche en déduisant la même direction de recherche d'une manière différente. Nous introduisons les notations suivantes :

$$v = \sqrt{\frac{xz}{\mu}}, \quad d = \sqrt{\frac{x}{z}},$$

et les directions de recherche réduites d_x et d_z suivantes :

$$d_x = \frac{v \Delta x}{x}, \quad d_z = \frac{v \Delta z}{z}. \quad (2.2.6)$$

Le système (2.2.4) peut s'écrire comme suit :

$$\begin{cases} \bar{A}^t \Delta y + d_z - \bar{Q} d_x = 0, \\ \bar{A} d_x = 0, \\ d_x + d_z = v^{-1} - v, \end{cases} \quad (2.2.7)$$

où $\bar{A} = \frac{1}{\sqrt{\mu}}AD$ et $\bar{Q} = DQD$ avec $D = \text{diag}(d)$.

Nous pouvons conclure que l'itéré $(x, y, z) \in F^0$ coïncide avec le centre $(x(\mu), y(\mu), z(\mu))$ si et seulement si $v = e$.

Remarque 2.2.1. Soit $(d_x, \Delta y, d_z)$ une solution du système (2.2.7) et comme \bar{Q} est une matrice semi-définie positive, on note que

$$(d_x)^t d_z = (d_x)^t (\bar{Q}d_x - \bar{A}^t \Delta y) = (d_x)^t \bar{Q}d_x \geq 0,$$

cette dernière inégalité montre que d_x et d_z ne sont pas des directions orthogonales contrairement à l'OL.

Une remarque cruciale est que le côté droit de la troisième équation en (2.2.7) qui est appelée l'équation de centralité réduite égal moins le gradient de la fonction barrière logarithmique classique $\Phi : \mathfrak{R}_{++}^n \rightarrow \mathfrak{R}_+$ qui est définie comme suit :

$$\Phi(v) = \Phi(x, z; \mu) = \sum_{i=1}^n \psi(v_i), \quad (2.2.8)$$

où

$$\psi(v_i) = \frac{v_i^2 - 1}{2} - \log v_i, v \in \mathfrak{R}_{++}^n. \quad (2.2.9)$$

De plus, nous appelons ψ la fonction noyau de la fonction barrière logarithmique classique $\Phi(v)$. Ainsi, le système (2.2.7) devient :

$$\begin{cases} \bar{A}^t \Delta y + d_z - \bar{Q}d_x = 0, \\ \bar{A}d_x = 0, \\ d_x + d_z = -\nabla \Phi(v). \end{cases} \quad (2.2.10)$$

Notons que la matrice $\nabla^2 \Phi(v) = \text{diag}(e + v^{-2})$ est définie positive, donc $\Phi(v)$ est strictement convexe. De plus, puisque $\nabla \Phi(e) = 0$, il s'ensuit que $\Phi(v)$ atteint sa valeur minimale à $v = e$, avec $\Phi(e) = 0$.

Donc, la mesure de proximité basée sur la norme (1.3.2) devient

$$\delta(v) = \frac{1}{2} \|v^{-1} - v\| = \frac{1}{2} \|d_x + d_z\| = \frac{1}{2} \|\nabla \Phi(v)\|, v \in \mathfrak{R}_{++}^n. \quad (2.2.11)$$

2.3 Description algorithmique

On peut caractériser une étape de *MPIs* primales-duales basées sur une fonction noyau comme suit : en commençant par un point strictement réalisable (x^0, y^0, z^0) , un paramètre de précision $\varepsilon > 0$, un paramètre de seuil $\tau \geq 1$ et la fonction de proximité $\Phi(v)$ qui est une bonne proximité du μ -center $(x(\mu), y(\mu), z(\mu))$ pour une valeur fixée de μ à l'itération actuelle. Ensuite, nous diminuons le paramètre μ par un facteur $(1 - \theta)$ où $0 < \theta < 1$. Lorsque $\Phi(v) \leq \tau$, une solution approximative du μ -center est obtenue par (2.2.5) en utilisant fréquemment la méthode de Newton par la résolution du système (2.2.7) via (2.2.6) pour obtenir la direction $(\Delta x, \Delta y, \Delta z)$. Nous répétons cette procédure jusqu'à ce que nous obtenions le point où $n\mu \leq \varepsilon$. Dans ce cas, nous indiquons que les x et (y, z) actuels sont des solutions approximatives des problèmes (*PQC*) et (*DQC*), respectivement. Donc, l'algorithme de *MPIs* basées sur une fonction noyau est constitué de deux parties : les itérations internes et externes. Chaque itération externe comprend une mise à jour du paramètre μ et une séquence d'itérations internes (une ou plusieurs). La forme générique de l'algorithme est donnée comme suit :

Algorithme 5 : Algorithme générique des *MPIs* primales-duales pour la programmation quadratique convexe

Données :

une fonction de proximité $\Phi(v)$;

un paramètre de précision $\varepsilon > 0$;

un paramètre de seuil $\tau \geq 1$;

un paramètre de mise à jour (de barrière) fixé $\theta, 0 < \theta < 1$;

un point strictement réalisable (x^0, y^0, z^0) de (PQC) et (DQC) et $\mu^0 = 1$;

Initialisation

$$x = x^0, y = y^0, z = z^0, \mu = \mu^0, v = \sqrt{\frac{xz}{\mu}};$$

Tant que $n\mu \geq \varepsilon$ **faire**

Début (itération externe)

$$\mu = (1 - \theta)\mu;$$

Tant que $\Phi(v) > \tau$ **faire**

Début (itération interne)

résoudre le système (2.2.7) via (2.2.6), pour obtenir la direction $(\Delta x, \Delta y, \Delta z)$;

choisir un pas convenable α et calculer :

$$x = x + \alpha\Delta x; y = y + \alpha\Delta y; z = z + \alpha\Delta z;$$

$$v = \sqrt{\frac{xz}{\mu}};$$

Fin

Fin

Fin Algorithme

2.4 Analyse de l'algorithme

Dans cette section, nous donnons une description de l'analyse des *MPIs* basées sur une fonction noyau pour résoudre les problèmes de (PQC). Tout d'abord, nous étudions le comportement de croissance de la fonction barrière $\Phi(v)$ au cours d'une itération externe. Ensuite, nous choisissons un pas de déplacement α , qui doit être pris de telle sorte que la proximité des itérés avec le μ -centre actuel s'améliore suffisamment. Enfin, nous étudions la décroissance de la fonction barrière $\Phi(v)$ au cours d'une itération interne.

2.4.1 Borne supérieure de la fonction barrière $\Phi(v)$ pendant une itération externe

Notons qu'au début de chaque itération externe de l'algorithme, juste avant la mise à jour de μ , nous avons $\Phi(v) \leq \tau$. Par la mise à jour de μ , le vecteur v se divise par $\sqrt{1-\theta}$ i.e., le vecteur v est mis à jour par la relation suivante $v^+ = \frac{v}{\sqrt{1-\theta}}$, ce qui conduit généralement à une augmentation de la valeur de $\Phi(v)$. Ainsi, au cours de l'algorithme, les plus grandes valeurs de $\Phi(v)$ apparaissent justes après les mises à jour de μ . Dans ce paragraphe, nous trouvons donc une estimation de l'effet d'une mise à jour de μ sur la valeur de $\Phi(v)$. En d'autres termes, nous voulons obtenir une borne supérieure pour $\Phi(\beta v)$ avec $\beta = \frac{1}{\sqrt{1-\theta}}$ en terme de $\Phi(v)$.

Il apparaîtra clairement que dans l'analyse de l'algorithme, certaines fonctions inverses liées aux fonctions noyaux jouent un rôle essentiel. Nous introduisons ici ces fonctions inverses.

$$\begin{aligned} \sigma : [0, +\infty[&\rightarrow [1, +\infty[\text{ la fonction inverse de } \psi(t), \forall t \geq 1, \\ \rho : [0, +\infty[&\rightarrow]0, 1] \text{ la fonction inverse de } -\frac{1}{2}\psi'(t), \forall t \in]0, 1]. \end{aligned}$$

Dans la suite, on suppose que la fonction noyau est qualifiée.

Nous offrons un théorème important qui est valable pour toutes les fonctions noyaux qui satisfont la condition de qualification **(C.g.e)**.

Théorème 2.4.1. ([28]) Soit $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de $\psi(t)$ pour $t \geq 1$. Alors, nous avons :

$$\Phi(\beta v) \leq n\psi\left(\beta\sigma\left(\frac{\Phi(v)}{n}\right)\right), v \in \mathbb{R}_{++}^n, \beta > 1.$$

Démonstration. Pour tout $\beta > 1$, nous considérons le problème de maximisation suivant :

$$\max_v \{\Phi(\beta v) : \Phi(v) = y, y \geq 0\},$$

les conditions d'optimalité de premier ordre pour ce problème sont :

$$\beta\psi'(\beta v_i) = \kappa\psi'(v_i), \quad i = 1, \dots, n, \quad (2.4.1)$$

où κ représente le multiplicateur de Lagrange.

Puisque $\psi'(1) = 0$ et $\beta\psi'(\beta) > 0$, nous devons avoir $v_i \neq 1$ pour tous les i . On peut aussi supposer que $v_i > 1$ pour tous les i .

Soit $\psi(v_i) = y_i$ une équation admet deux solutions $v_i^{(1)} > 1$ et $v_i^{(2)} < 1$. La conséquence du Lemme 2.1.8 est que nous avons $\psi(\beta v_i^{(2)}) \leq \psi(\beta v_i^{(1)})$, $\forall \beta > 1$. Comme nous maximisons $\Phi(\beta v)$, il s'ensuit que nous pouvons supposer que $v_i = v_i^{(1)} > 1$ pour tous les i . Notons que (2.4.1) implique alors que $\beta\psi'(\beta v_i) > 0$ et $\psi'(v_i) > 0$, d'où également $\kappa > 0$. En définissant maintenant :

$$\begin{aligned} h(t) &= \frac{\psi'(t)}{\psi'(\beta t)} = \frac{\beta}{\kappa}, t > 1, \\ h'(t) &= \frac{\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t)}{(\psi'(\beta t))^2}. \end{aligned}$$

En utilisant que $\psi(t)$ satisfait la condition de qualification **(C.g.e)**, nous trouvons que $h'(t) > 0$ pour $t > 1$ et $\beta > 1$. Ainsi, nous avons prouvé que $h(t)$ est strictement croissante. Il en résulte donc que tous les v_i sont égaux. En mettant $v_i = t > 1$, pour tout $i = 1, \dots, n$, on déduit de $\Phi(v) = y$ que $n\psi(t) = y$. Cela implique que $t = \sigma\left(\frac{\Phi(v)}{n}\right)$.

Par conséquent, la valeur maximale que $\Phi(v)$ peut atteindre est donnée par

$$\Phi(\beta t) = n\psi(\beta t) = n\psi\left(\beta\sigma\left(\frac{y}{n}\right)\right) = n\psi\left(\beta\sigma\left(\frac{\Phi(v)}{n}\right)\right).$$

Ce qui termine la preuve ■

Corollaire 2.4.2. Pour tout vecteur positif v , si $\Phi(v) \leq \tau$ et $\beta > 1$, on a :

$$\Phi(\beta v) \leq \frac{n}{2}\psi''(1)\left(\beta\sigma\left(\frac{\tau}{n}\right) - 1\right)^2.$$

Démonstration. Pour tout vecteur positif v , si $\Phi(v) \leq \tau$ et $\beta > 1$ alors par le Théorème 2.4.1, nous avons :

$$\Phi(\beta v) \leq n\psi\left(\beta\sigma\left(\frac{\Phi(v)}{n}\right)\right).$$

Comme σ est une fonction croissante, on a :

$$\Phi(\beta v) \leq n\psi\left(\beta\sigma\left(\frac{\tau}{n}\right)\right),$$

puisque $\beta > 1$ et $\sigma\left(\frac{\tau}{n}\right) \geq 1$, en utilisant le Lemme 2.1.5, on obtient :

$$\Phi(\beta v) \leq \frac{n}{2}\psi''(1)\left(\beta\sigma\left(\frac{\tau}{n}\right) - 1\right)^2.$$

La preuve est ainsi faite. ■

Par le Corollaire 2.4.2 avec $\beta = \frac{1}{\sqrt{1-\theta}}$, nous avons :

$$\Phi(v^+) = \Phi\left(\frac{v}{\sqrt{1-\theta}}\right) \leq \frac{n}{2}\psi''(1)\left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2.$$

Nous notons par

$$\Phi_0 = \frac{n}{2}\psi''(1)\left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2, \quad (2.4.2)$$

la borne supérieure de $\Phi(v^+)$; la valeur de $\Phi(v)$ après la mise à jour de μ .

2.4.2 Détermination du pas de déplacement α et la décroissance de $\Phi(v)$ pendant une itération interne

Dans chaque itération interne, nous calculons d'abord la direction $(\Delta x, \Delta y, \Delta z)$ puis le pas de déplacement α . Rappelons qu'au cours d'une itération interne, le paramètre μ est fixé. Après un pas de déplacement, nous avons :

$$x^+ = x + \alpha\Delta x, \quad y^+ = y + \alpha\Delta y, \quad z^+ = z + \alpha\Delta z.$$

En utilisant (2.2.6), on obtient :

$$x^+ = x + \alpha\Delta x = x\left(e + \alpha\frac{\Delta x}{x}\right) = \frac{x}{v}(v + \alpha d_x),$$

et

$$z^+ = z + \alpha\Delta z = z\left(e + \alpha\frac{\Delta z}{z}\right) = \frac{z}{v}(v + \alpha d_z).$$

En utilisant $xz = \mu v^2$, donc nous obtenons :

$$v_+ = \sqrt{\frac{x^+ z^+}{\mu}} = \sqrt{(v + \alpha d_x)(v + \alpha d_z)}.$$

Pour μ fixé, nous définissons la différence de proximité entre le nouvel itéré et l'ancien comme suit :

$$f(\alpha) = \Phi(v_+) - \Phi(v) = \Phi\left(\sqrt{(v + \alpha d_x)(v + \alpha d_z)}\right) - \Phi(v).$$

Il est clair que $f(\alpha)$ n'est pas nécessairement convexe. Pour faciliter l'analyse, nous utilisons une borne supérieure convexe pour $f(\alpha)$. On obtient cette borne en utilisant que $\psi(t)$ est e-convexe (selon Lemme 2.1.2). Cela donne :

$$\Phi(v_+) = \Phi\left(\sqrt{(v + \alpha d_x)(v + \alpha d_z)}\right) \leq \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_z)).$$

Donc, nous avons :

$$f(\alpha) \leq f_1(\alpha) = \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_z)) - \Phi(v). \quad (2.4.3)$$

Il est clair que $f(0) = f_1(0) = 0$. On prend les deux premières dérivées de $f_1(\alpha)$ par rapport à α , on a :

$$f_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n \left(\psi'(v_i + \alpha d_{x_i}) d_{x_i} + \psi'(v_i + \alpha d_{z_i}) d_{z_i} \right),$$

$$f_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n \left(\psi''(v_i + \alpha d_{x_i}) (d_{x_i})^2 + \psi''(v_i + \alpha d_{z_i}) (d_{z_i})^2 \right). \quad (2.4.4)$$

où d_{x_i} et d_{z_i} désignent respectivement la $i^{\text{ème}}$ composante des vecteurs d_x et d_z .

En utilisant (2.2.10) et (2.2.11), nous avons :

$$f_1'(0) = \frac{1}{2} \langle \nabla \Phi(v), d_x + d_z \rangle = -\frac{1}{2} \langle \nabla \Phi(v), \nabla \Phi(v) \rangle = -2(\delta(v))^2. \quad (2.4.5)$$

Dans la suite, nous utilisons les notations abrégées suivantes :

$$v_{\min} = \min_{i=1, \dots, n} (v_i), \quad \delta = \delta(v) \text{ et } \Phi = \Phi(v).$$

Lemme 2.4.3. Soient $f_1(\alpha)$ définie dans (2.4.3) et δ défini dans (2.2.11). Alors, nous avons :

$$f_1''(\alpha) \leq 2\delta^2 \psi''(v_{\min} - 2\alpha\delta).$$

Démonstration. D'après la Remarque 2.2.1, on a $(d_x)^t d_z = (d_x)^t \bar{Q} d_x \geq 0$, cela implique que :

$$\begin{aligned} 4\delta^2 &= \|d_x + d_z\|^2 = \|d_x\|^2 + \|d_z\|^2 + 2(d_x)^t d_z, \\ &\geq \|d_x\|^2 + \|d_z\|^2, \end{aligned}$$

donc

$$\|d_x\| \leq 2\delta, \quad \|d_z\| \leq 2\delta.$$

On a :

$$v_i + \alpha d_{x_i} \geq v_{\min} - 2\alpha\delta, \quad v_i + \alpha d_{z_i} \geq v_{\min} - 2\alpha\delta, \quad \forall i = 1, \dots, n.$$

D'après la condition de qualification **(C.g.c)**, $\psi''(t)$ est strictement décroissante.

Donc d'après (2.4.4), on obtient :

$$\begin{aligned} f_1''(\alpha) &\leq \frac{1}{2} \psi''(v_{\min} - 2\alpha\delta) \sum_{i=1}^n \left((d_{x_i})^2 + (d_{z_i})^2 \right), \\ &\leq \frac{1}{2} \psi''(v_{\min} - 2\alpha\delta) \sum_{i=1}^n (d_{x_i} + d_{z_i})^2, \\ &= 2\delta^2 \psi''(v_{\min} - 2\alpha\delta). \end{aligned}$$

Ceci complète la preuve. ■

La stratégie pour le choix du pas de déplacement est une extension de celle de l'OL dans [7]. Le lemme suivant montre la décroissance de la fonction $f_1(\alpha)$.

Lemme 2.4.4. ([7]) Si le pas de déplacement α satisfait l'inégalité

$$-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \leq 2\delta, \tag{2.4.6}$$

alors

$$f_1'(\alpha) \leq 0.$$

Démonstration. En utilisant le Lemme 2.4.3 et (2.4.5), nous pouvons écrire :

$$\begin{aligned} f_1'(\alpha) &= f_1'(0) + \int_0^\alpha f_1''(\xi) d\xi, \\ &\leq -2\delta^2 + \int_0^\alpha 2\delta^2 \psi''(v_{\min} - 2\xi\delta) d\xi, \\ &= -2\delta^2 - \delta \int_0^\alpha \psi''(v_{\min} - 2\xi\delta) d(v_{\min} - 2\xi\delta), \\ &= -2\delta^2 + \delta \left(-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \right), \end{aligned}$$

puisque $-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \leq 2\delta$, alors $f_1'(\alpha) \leq 0$.

Ce qui conclut la preuve du lemme. ■

Lemme 2.4.5. ([7]) La plus grande valeur du pas α vérifiant (2.4.6) est donnée par :

$$\bar{\alpha} = \frac{(\rho(\delta) - \rho(2\delta))}{2\delta}.$$

Démonstration. Nous cherchons une grande valeur possible de α vérifiant (2.4.6).

Puisque $\psi''(t)$ est décroissante alors on a :

$$-\psi''(v_{\min} - 2\alpha\delta) + \psi''(v_{\min}) \leq 0,$$

pour v_{\min}, α et δ sont fixés.

La dérivée par rapport à α de l'expression à gauche dans (2.4.6) est égale à

$$2\delta\psi''(v_{\min} - 2\alpha\delta) \geq 0.$$

Donc la valeur maximale pour l'expression à gauche dans (2.4.6) par rapport à v_{\min} est atteinte si v_{\min} atteint sa valeur minimale. Pour $0 < v_{\min} \leq 1$, on a :

$$\delta(v) = \frac{1}{2} \|\nabla\Phi(v)\| \geq \frac{1}{2} |\psi'(v_{\min})| \geq -\frac{1}{2}\psi'(v_{\min}),$$

puisque ρ est une fonction strictement décroissante, on a $\rho(\delta) \leq v_{\min}$, nous prenons :

$$\rho(\delta) = v_{\min}, \quad (2.4.7)$$

donc

$$-2\delta = \psi'(v_{\min}). \quad (2.4.8)$$

On remplaçant (2.4.8) dans (2.4.6) on obtient :

$$-\psi'(v_{\min} - 2\alpha\delta) \leq 4\delta,$$

donc nous prenons :

$$v_{\min} - 2\alpha\delta = \rho(2\delta). \quad (2.4.9)$$

D'après (2.4.7) et (2.4.9), on a :

$$\rho(\delta) = v_{\min} \quad \text{et} \quad v_{\min} - 2\alpha\delta = \rho(2\delta),$$

donc

$$\bar{\alpha} = \alpha = \frac{(\rho(\delta) - \rho(2\delta))}{2\delta}.$$

Ce qui termine la preuve du lemme. ■

Lemme 2.4.6. ([7]) Soit $\bar{\alpha}$ défini dans Lemme 2.4.5, on a :

$$\bar{\alpha} \geq \frac{1}{\psi''(\rho(2\delta))}.$$

Démonstration. Par la définition du ρ ,

$$-2\delta = \psi'(\rho(\delta)),$$

La dérivée obtenue par rapport à δ est la suivante :

$$-2 = \rho'(\delta) \psi''(\rho(\delta)),$$

ce qui implique que

$$\rho'(\delta) = -\frac{2}{\psi''(\rho(\delta))} < 0,$$

Par conséquent, ρ est une fonction décroissante. Une conséquence immédiate de dernière égalité et Lemme 2.4.5 est :

$$\begin{aligned} \bar{\alpha} &= \frac{(\rho(\delta) - \rho(2\delta))}{2\delta}, \\ &= \frac{1}{2\delta} \int_{2\delta}^{\delta} \rho'(\varsigma) d\varsigma, \\ &= \frac{1}{\delta} \int_{2\delta}^{\delta} -\frac{1}{\psi''(\rho(\varsigma))} d\varsigma. \end{aligned}$$

A cause de la condition de qualification **(C.g.c)**, $\psi''(t)$ est strictement décroissante sur $[\delta, 2\delta]$. Par conséquent,

$$\begin{aligned} \bar{\alpha} &\geq -\frac{1}{\delta} \frac{1}{\psi''(\rho(2\delta))} \int_{2\delta}^{\delta} d\varsigma, \\ &= \frac{1}{\psi''(\rho(2\delta))}. \end{aligned}$$

Ce qui conclut la preuve du lemme. ■

On note par

$$\tilde{\alpha} = \frac{1}{\psi''(\rho(2\delta))}, \tag{2.4.10}$$

le pas de déplacement tel que $\bar{\alpha} \geq \tilde{\alpha}$. Nous avons besoin d'un résultat technique simple suivant.

Lemme 2.4.7. ([61]) *Soit $g(t) \in C^2(\mathfrak{R}_+)$ une fonction convexe avec $g(0) = 0, g'(0) < 0$ et $g(t)$ atteint son minimum global à $t^* > 0$. Si $g''(t)$ est décroissante pour $t \in [0, t^*]$, alors :*

$$g(t) \leq t \left[\frac{g'(0)}{2} \right].$$

Lemme 2.4.8. ([7]) *Si le pas de déplacement α vérifie $\alpha \leq \bar{\alpha}$, alors*

$$f(\alpha) \leq -\alpha\delta^2.$$

Démonstration. Soit $g(\alpha)$ définie comme suit :

$$g(\alpha) = -2\alpha\delta^2 + \frac{1}{2}\psi(v_{\min} - 2\alpha\delta) + \alpha\delta\psi'(v_{\min}) - \frac{1}{2}\psi(v_{\min}),$$

alors, on a :

$$g(0) = f_1(0) = 0; \quad g'(0) = f_1'(0) = -2\delta^2,$$

et

$$\begin{aligned} g''(\alpha) &= 2\delta^2\psi''(v_{\min} - 2\alpha\delta), \\ g'''(\alpha) &= -4\delta^3\psi'''(v_{\min} - 2\alpha\delta). \end{aligned}$$

À cause du Lemme 2.4.3, on a $f_1''(\alpha) \leq g''(\alpha)$. Par conséquent,

$$\begin{aligned} f_1'(\alpha) &\leq g'(\alpha), \\ f_1(\alpha) &\leq g(\alpha). \end{aligned}$$

En prenant $\alpha \leq \bar{\alpha}$, avec $\bar{\alpha}$ défini dans le Lemme 2.4.5, on obtient :

$$\begin{aligned} g'(\alpha) &= g'(0) + \int_0^\alpha g''(\xi) d\xi, \\ &= -2\delta^2 + 2\delta^2 \int_0^\alpha \psi''(v_{\min} - 2\delta\xi) d\xi, \\ &= -2\delta^2 - \delta \int_0^\alpha \psi''(v_{\min} - 2\delta\xi) d(v_{\min} - 2\delta\xi), \\ &= -2\delta^2 + \delta \left(-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \right), \\ &\leq 0. \end{aligned}$$

Puisque $g''(\alpha)$ est une fonction décroissante, en utilisant le Lemme 2.4.7, on peut écrire

$$f_1(\alpha) \leq g(\alpha) \leq \frac{\alpha g'(0)}{2} = -\alpha\delta^2,$$

comme $f(\alpha) \leq f_1(\alpha)$, donc on a :

$$f(\alpha) \leq -\alpha\delta^2.$$

Ce qui termine la preuve du lemme. ■

En combinant les résultats des Lemmes 2.4.6 et 2.4.8, on obtient le théorème suivant qui exprime la décroissance de la fonction barrière pendant une itération interne.

Théorème 2.4.9. ([28]) *Si $\tilde{\alpha}$ est le pas de déplacement qui est défini dans (2.4.10), on a :*

$$f(\tilde{\alpha}) \leq -\frac{\delta^2}{\psi''(\rho(2\delta))}.$$

2.5 Complexité algorithmique

Dans cette section, nous présentons les résultats généraux de complexité pour les méthodes de points intérieurs primales-duales basées sur les fonctions noyaux qui satisfont certaines conditions de qualification.

Après la mise à jour de μ et selon (2.4.2), on a :

$$\Phi(v^+) \leq \Phi_0 = \frac{n}{2}\psi''(1) \left(\frac{\sigma(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1 \right)^2,$$

c'est la première valeur de Φ dans l'itération externe.

Nous avons besoin de compter le nombre d'itérations internes nécessaires pour revenir à la situation $\Phi \leq \tau$. Les valeurs suivantes dans la même itération externe sont désignées par Φ_k , $k = 1, 2, \dots, K$, où K représente le nombre total d'itérations internes dans l'itération externe. D'autre part, selon le Théorème 2.4.9, nous avons :

$$f(\alpha) = \Phi_{k+1} - \Phi_k \leq f(\tilde{\alpha}) \leq -\frac{\delta^2}{\psi''(\rho(2\delta))},$$

alors on suppose qu'ils existent $\bar{\kappa} > 0$ et $\gamma \in]0, 1]$, tels que

$$\Phi_{k+1} - \Phi_k \leq -\bar{\kappa}\Phi_k^{1-\gamma} \leq -\bar{\kappa}(\Phi_k - \tau)^{1-\gamma},$$

alors

$$(\Phi_{k+1} - \tau) \leq (\Phi_k - \tau) - \bar{\kappa}(\Phi_k - \tau)^{1-\gamma}.$$

Lemme 2.5.1. ([61]) Soit t_0, t_1, \dots, t_K une suite des nombres positifs vérifiant :

$$t_{k+1} \leq t_k - \bar{\kappa}t_k^{1-\gamma}, k = 0, 1, \dots, K - 1,$$

tels que $\bar{\kappa} > 0$ et $\gamma \in]0, 1]$, alors :

$$K \leq \frac{(t_0)^\gamma}{\bar{\kappa}\gamma}.$$

En utilisant le Lemme 2.5.1 pour $t_k = \Phi_k - \tau > 0$ alors K le nombre d'itérations internes dans chaque itération externe est donnée par :

$$K \leq \frac{(\Phi_0 - \tau)^\gamma}{\bar{\kappa}\gamma} \leq \frac{(\Phi_0)^\gamma}{\bar{\kappa}\gamma} = \frac{\left(\frac{n}{2}\psi''(1)\left(\frac{\sigma(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1\right)^2\right)^\gamma}{\bar{\kappa}\gamma}.$$

La détermination du nombre total d'itérations pour trouver une solution primale-duale optimale nécessite le calcul du nombre d'itérations externes.

Théorème 2.5.2. Soit k le nombre d'itérations externes nécessaires pour trouver une solution optimale primale-duale approchée avec une précision $\varepsilon > 0$, alors on a :

$$k \geq \frac{1}{\theta} \log \frac{n}{\varepsilon}.$$

Démonstration. Si x^k est une solution optimale primale-duale approchée du problème alors on a $n\mu^k \leq \varepsilon$. Et on a :

$$\mu^k = (1 - \theta)^k \mu^0 = (1 - \theta)^k, \mu^0 = 1,$$

donc on trouve :

$$n(1 - \theta)^k \leq \varepsilon \implies k \log(1 - \theta) \leq \log \frac{\varepsilon}{n}.$$

Puisque $\log(1 - \theta) \leq -\theta$, on a :

$$k\theta \geq \log \frac{n}{\varepsilon} \implies k \geq \frac{1}{\theta} \log \frac{n}{\varepsilon}.$$

Ce qui conclut la preuve du lemme. ■

En multipliant le nombre d'itérations externes par le nombre d'itérations internes, on obtient une borne supérieure du nombre total d'itérations, à savoir :

$$\frac{(\Phi_0)^\gamma}{\theta \bar{\kappa}^\gamma} \log \frac{n}{\varepsilon} = \frac{\left(\frac{n}{2} \psi''(1) \left(\frac{\sigma(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1 \right)^2 \right)^\gamma}{\theta \bar{\kappa}^\gamma} \log \frac{n}{\varepsilon}. \quad (2.5.1)$$

Notre objectif est de calculer le nombre d'itérations de *MPIs* à long et court-pas basées sur les fonctions noyaux. Les méthodes à long-pas sont caractérisées par $\tau = O(n)$ et $\theta = \Theta(1)$ et les méthodes à court-pas sont caractérisées par $\tau = O(1)$ et $\theta = \Theta(\frac{1}{\sqrt{n}})$.

Conclusion 2.5.3. *Dans ce chapitre, nous avons présenté les bases théoriques et algorithmiques de la méthode de trajectoire centrale basée sur une fonction noyau qualifiée générale pour la programmation quadratique convexe. L'avantage de cette approche est de démarrer l'algorithme d'un point strictement réalisable non nécessairement centré c.-à-d. vérifiant seulement la condition (CPI). L'inconvénient de cette méthode est la complexité algorithmique; malgré qu'elle est polynomiale; elle est égale au produit du nombre d'itérations externes et du nombre d'itérations internes. Le but du prochain chapitre est d'améliorer la complexité de l'algorithme par la réduction du nombre d'itérations internes. D'après (2.5.1), cela dépend de la fonction noyau, sa forme et ses propriétés. Donc, nous allons proposer deux nouvelles fonctions noyaux par d'autre façon et deux nouvelles directions de déplacement capables d'améliorer le comportement de l'algorithme en question.*

CHAPITRE 3

MÉTHODE DE POINTS INTÉRIEURS DE TYPE TRAJECTOIRE CENTRALE BASÉE SUR DES NOUVELLES FONCTIONS NOYAUX POUR (PQC)

L'amélioration du comportement de l'algorithme repose sur l'idée de remplacement de la fonction barrière logarithmique classique (2.2.8) par une nouvelle fonction barrière $\Phi_{new}(v)$ et le remplacement de la fonction noyau logarithmique classique (2.2.9) par une nouvelle fonction noyau ψ_{new} . Donc, le système de Newton (2.2.10) devient :

$$\begin{cases} \bar{A}^t \Delta y + d_z - \bar{Q} d_x = 0, \\ \bar{A} d_x = 0, \\ d_x + d_z = -\nabla \Phi_{new}(v). \end{cases}$$

où $\Phi_{new}(v) = \sum_{i=1}^n \psi_{new}(v_i)$.

La mesure de proximité (2.2.11) qui mesure la distance entre le μ -centre et l'itéré courant devient :

$$\delta(v) = \frac{1}{2} \|d_x + d_z\| = \frac{1}{2} \|\nabla \Phi_{new}(v)\|, v \in \mathfrak{R}_{++}^n.$$

Il existe plusieurs types des termes barrières des fonctions noyaux pour l'OL, on cite par exemple : logarithmique [16, 62], exponentiel [7, 14], polynomial (de Laurent) et trigonométrique [15, 17, 29]. On s'intéresse dans cette thèse aux fonctions noyaux avec terme barrière exponentiel et polynomial (de Laurent). Dans les tableaux 1 et 2, on cite quelques fonctions noyaux de ces deux types.

ψ_i	Formule	Court-pas	Long-pas	Référence
1	$\frac{t^2-1}{2} + \frac{e^{\frac{1}{t}}-e}{e}$	$O(\sqrt{n} \log n)$	$O\left(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon}\right)$	[7]
2	$\frac{t^2-1}{2} + \int_1^t e^{\frac{1}{x}-1} dx$	$O(\sqrt{n} \log n)$	$O\left(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon}\right)$	[7]
3	$\frac{t^2-1}{2} + e^{\frac{1}{t}-1} - 1$	$O(\sqrt{n} \log n)$	$O\left(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon}\right)$	[7]
4	$p \frac{t^2-1}{2} + e^{p(\frac{1}{t}-1)} - 1, p > 0$	$O(\sqrt{n} \log n)$	$O\left(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon}\right)$	[14]
N	$\frac{p(t^2-1)}{2} - \int_1^t p e^{p(\frac{1}{x}-1)} dx, p \geq 1$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O\left(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon}\right)$	[18]

Tableau 1 : Fonctions noyaux avec terme barrière exponentiel et leur complexité algorithmique pour l'OL.

ψ_i	Formule	Court-pas	Long-pas	Référence
1	$\frac{1}{2} \left(t - \frac{1}{t}\right)^2$	$O(\sqrt{n} \log n)$	$O\left(n^{\frac{2}{3}} \log \frac{n}{\varepsilon}\right)$	[59]
2	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}, q > 1$	$O(q^2 \sqrt{n} \log n)$	$O\left(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon}\right)$	[60]
3	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q(q-1)}$ $-\frac{q-1}{q}(t-1), q > 1$	$O(q\sqrt{n} \log n)$	$O\left(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon}\right)$	[61]
4	$t-1 + \frac{t^{1-q}-1}{q-1}, q > 1$	$O(q^2 \sqrt{n} \log n)$	$O(qn \log \frac{n}{\varepsilon})$	[7],[9]
B	$t^2 - t + \frac{t^{-(p+1)}-1}{p+1}, p > 0$	$O(p^2 \sqrt{n} \log \frac{n}{\varepsilon}),$ si $p \geq 1.$ $O(\sqrt{n} \log \frac{n}{\varepsilon}),$ si $p \in]0, 1].$	$O\left(pn^{\frac{p+3}{2(p+2)}} \log \frac{n}{\varepsilon}\right),$ si $p > 1.$ $O(\sqrt{n} \log n \log \frac{n}{\varepsilon}),$ si $p = \frac{\log n}{2} - 2.$ $O\left(n^{\frac{3}{4}} \log \frac{n}{\varepsilon}\right),$ si $p \in]0, 1].$	[19]

Tableau 2 : Fonctions noyaux avec terme barrière polynomial (de Laurent) et leur complexité algorithmique pour l'OL.

Dans les deux sections suivantes, on va présenter deux nouvelles fonctions noyaux paramétrées. La première fonction $\psi_{new} = \psi_N$ avec terme barrière exponentiel généralise celle donnée par Bai et al. [7] lorsque $p = 1$, à savoir ψ_2 dans le tableau 1. La deuxième fonction noyau $\psi_{new} = \psi_B$ avec terme barrière polynomial (de Laurent) pour améliorer de manière significative les résultats théoriques et pratiques obtenus par Peng et al. dans [60] et Bai et al. dans [9].

3.1 Nouvelle fonction noyau avec terme barrière exponentiel

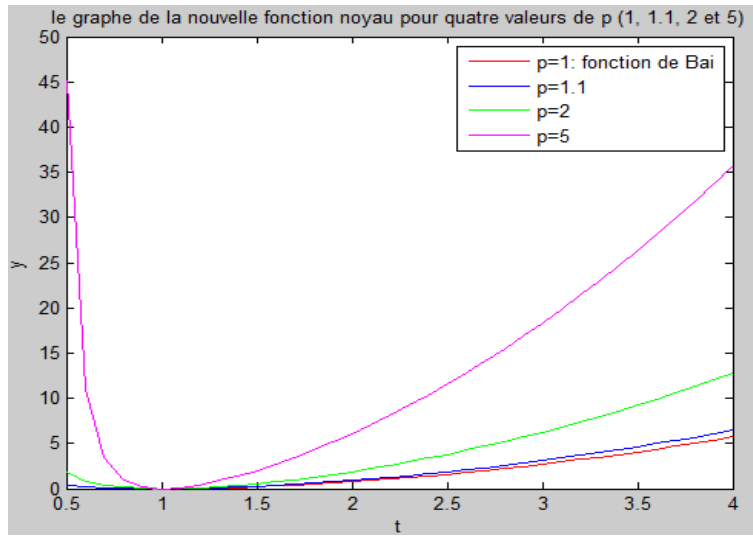
Cette section est consacrée à introduire notre nouvelle fonction noyau paramétrée avec un terme barrière exponentiel et à développer certaines propriétés utiles qui sont requises dans l'analyse de l'algorithme. Le contenu de cette partie a fait l'objet de notre premier article [18].

Notre nouvelle fonction noyau ψ_N est définie comme suit :

$$\psi_N(t) = \frac{p(t^2 - 1)}{2} - \int_1^t p e^{p(\frac{1}{x}-1)} dx, \quad (3.1.1)$$

où le paramètre $p \geq 1$.

Pour étudier le comportement de cette nouvelle fonction noyau au voisinage du point minimum, le graphe de la fonction ψ_N est tracé pour quatre valeurs différentes du paramètre $p = \{1, 1.1, 2, 5\}$ sur la figure suivante :



À partir de cette figure, nous constatons que le comportement de croissance de ψ_N pour $p = 1$ et 1.1 est moins rapide que pour $p = 2$ et $p = 5$ lorsque $t \rightarrow 0^+$ et $t \rightarrow +\infty$.

Nous avons besoin de ses trois premières dérivées suivantes :

$$\psi'_N(t) = pt - p e^{p(\frac{1}{t}-1)}, \quad (3.1.2)$$

$$\psi''_N(t) = p + \frac{p^2}{t^2} e^{p(\frac{1}{t}-1)}, \quad (3.1.3)$$

$$\psi'''_N(t) = -e^{p(\frac{1}{t}-1)} \left(\frac{2p^2}{t^3} + \frac{p^3}{t^4} \right). \quad (3.1.4)$$

Par évidence, nous avons :

$$\psi''_N(t) > p > 0, \quad (3.1.5)$$

$$\psi_N(1) = \psi'_N(1) = 0. \quad (3.1.6)$$

Il reste à montrer que $\psi_N(t)$ est une fonction barrière pour qu'elle soit une fonction noyau. D'une part :

$$\lim_{t \rightarrow 0^+} \psi_N(t) = \lim_{t \rightarrow 0^+} \left(\frac{p(t^2 - 1)}{2} - \int_1^t p e^{p(\frac{1}{x}-1)} dx \right) = \frac{-p}{2} + \lim_{t \rightarrow 0^+} \int_t^1 p e^{p(\frac{1}{x}-1)} dx.$$

Puisque $0 < x \leq 1$ et $p > 0$, nous avons :

$$\begin{aligned} e^{p(\frac{1}{x}-1)} &> p \left(\frac{1}{x} - 1 \right), \\ \Rightarrow \int_t^1 p e^{p(\frac{1}{x}-1)} dx &> \int_t^1 p^2 \left(\frac{1}{x} - 1 \right) dx, \\ \Rightarrow \int_t^1 p e^{p(\frac{1}{x}-1)} dx &> p^2 (-\log t - 1 + t), \\ \Rightarrow \lim_{t \rightarrow 0^+} \psi_N(t) &> \frac{-p}{2} + \lim_{t \rightarrow 0^+} p^2 (-\log t - 1 + t), \end{aligned}$$

alors

$$\lim_{t \rightarrow 0^+} \psi_N(t) = +\infty.$$

D'autre part :

$$\lim_{t \rightarrow +\infty} \psi_N(t) = \lim_{t \rightarrow +\infty} \left(\frac{p(t^2 - 1)}{2} - \int_1^t p e^{p(\frac{1}{x}-1)} dx \right) = \lim_{t \rightarrow +\infty} \int_1^t p \left(x - e^{p(\frac{1}{x}-1)} \right) dx.$$

Puisque $1 \leq x < +\infty$ et $p > 0$, nous avons :

$$\begin{aligned} e^{p(\frac{1}{x}-1)} &\leq 1, \\ \Rightarrow -e^{p(\frac{1}{x}-1)} &\geq -1, \\ \Rightarrow x - e^{p(\frac{1}{x}-1)} &\geq x - 1, \\ \Rightarrow p \left(x - e^{p(\frac{1}{x}-1)} \right) &\geq p(x - 1), \\ \Rightarrow \int_1^t p \left(x - e^{p(\frac{1}{x}-1)} \right) dx &\geq \int_1^t p(x - 1) dx, \\ \Rightarrow \psi_N(t) &\geq p \left[\frac{x^2}{2} - x \right]_1^t, \\ \Rightarrow \psi_N(t) &\geq p \left(\frac{t^2}{2} - t + \frac{1}{2} \right), \\ \Rightarrow \lim_{t \rightarrow +\infty} \psi_N(t) &\geq \lim_{t \rightarrow +\infty} p \left(\frac{t^2}{2} - t + \frac{1}{2} \right), \end{aligned}$$

donc

$$\lim_{t \rightarrow +\infty} \psi_N(t) = +\infty.$$

3.1.1 Propriétés de la nouvelle fonction noyau

Quelques propriétés d'éligibilité de notre nouvelle fonction noyau ψ_N qui est définie dans (3.1.1), sont présentées dans le lemme suivant.

Lemme 3.1.1. *Soit $\psi_N(t)$ définie dans (3.1.1). Alors, nous avons :*

$$\psi_N'''(t) < 0, \forall t > 0, \quad (3.1.7)$$

$$t\psi_N''(t) - \psi_N'(t) > 0, \forall t > 0, \quad (3.1.8)$$

$$t\psi_N''(t) + \psi_N'(t) > 0, \forall t > 0, \quad (3.1.9)$$

$$\psi_N''(t)\psi_N'(\beta t) - \beta\psi_N'(t)\psi_N''(\beta t) > 0, \forall \beta > 1, \forall t > 1. \quad (3.1.10)$$

Démonstration. Pour montrer (3.1.7), en utilisant (3.1.4), on obtient :

$$\psi_N'''(t) < 0, \forall t > 0.$$

Pour montrer (3.1.8) et (3.1.9), en utilisant (3.1.2), (3.1.3) et la positivité de t, p et $e^{p(\frac{1}{t}-1)}$, on obtient :

$$\begin{aligned} t\psi_N''(t) - \psi_N'(t) &= t \left(p + \frac{p^2}{t^2} e^{p(\frac{1}{t}-1)} \right) - pt + pe^{p(\frac{1}{t}-1)}, \\ &= e^{p(\frac{1}{t}-1)} \left(\frac{p^2}{t} + p \right) > 0, \forall t > 0, \end{aligned}$$

et

$$\begin{aligned} t\psi_N''(t) + \psi_N'(t) &= t \left(p + \frac{p^2}{t^2} e^{p(\frac{1}{t}-1)} \right) + pt - pe^{p(\frac{1}{t}-1)}, \\ &= 2pt + e^{p(\frac{1}{t}-1)} \left(\frac{p^2}{t} - p \right). \end{aligned}$$

D'un part, si $0 < t < 1$ alors on a :

$$\begin{aligned} t\psi_N''(t) + \psi_N'(t) &= 2pt + e^{p(\frac{1}{t}-1)} \left(\frac{p^2}{t} - p \right), \\ &> p^2 - p > 0, \forall p \geq 1. \end{aligned}$$

D'autre part, si $t \geq 1$ alors on a :

$$\begin{aligned} t\psi_N''(t) + \psi_N'(t) &= 2pt + e^{p(\frac{1}{t}-1)} \left(\frac{p^2}{t} - p \right), \\ &> 2p - e^{p(\frac{1}{t}-1)}p = \left(2 - e^{p(\frac{1}{t}-1)} \right) p, \\ &> p > 0. \end{aligned}$$

Comme ψ_N vérifie (3.1.7) et (3.1.8) et d'après Lemme 2.1.7, on a :

$$\psi_N''(t) \psi_N'(\beta t) - \beta \psi_N'(t) \psi_N''(\beta t) > 0, \forall \beta > 1, \forall t > 1.$$

Ce qui conclut la preuve du lemme. ■

Dans la suite, nous présentons quelques résultats techniques de notre nouvelle fonction noyau nécessaires dans l'analyse.

Lemme 3.1.2. Pour $\psi_N(t)$, nous avons :

$$\frac{p}{2}(t-1)^2 < \psi_N(t) < \frac{1}{2p} \left(\psi_N'(t) \right)^2, \forall t > 0. \quad (3.1.11)$$

$$\psi_N(t) \leq \frac{1}{2} \psi_N''(1) (t-1)^2 = \frac{p^2+p}{2} (t-1)^2, \forall t \geq 1. \quad (3.1.12)$$

Démonstration. Pour montrer (3.1.11), en utilisant (3.1.5), nous avons :

$$\psi_N(t) = \int_1^t \int_1^x \psi_N''(y) dy dx > \int_1^t \int_1^x p dy dx = \frac{p}{2}(t-1)^2,$$

et

$$\begin{aligned} \psi_N(t) &= \int_1^t \int_1^x \psi_N''(y) dy dx, \\ &< \int_1^t \int_1^x \psi_N''(y) \frac{\psi_N''(x)}{p} dy dx, \\ &< \frac{1}{p} \int_1^t \psi_N''(x) \psi_N'(x) dx = \frac{1}{2p} \left(\psi_N'(t) \right)^2. \end{aligned}$$

Pour montrer (3.1.12), puisque ψ_N vérifie (3.1.6), (3.1.7) et $\psi_N''(1) = p^2 + p$, en

utilisant le développement de Taylor, nous avons pour certain ζ ($1 \leq \zeta \leq t$) :

$$\begin{aligned}\psi_N(t) &= \psi_N(1) + \psi'_N(1)(t-1) + \frac{1}{2}\psi''_N(1)(t-1)^2 + \frac{1}{6}\psi'''_N(\zeta)(\zeta-1)^3, \\ &\leq \frac{1}{2}\psi''_N(1)(t-1)^2 = \frac{p^2+p}{2}(t-1)^2.\end{aligned}$$

Ce qui termine la preuve du lemme. ■

Lemme 3.1.3. Soient $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de $\psi_N(t)$ et $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de $-\frac{1}{2}\psi'_N(t)$. Pour $\psi_N(t)$, on a :

$$1 + \sqrt{\frac{2}{p^2+p}s} \leq \sigma(s) < 1 + \sqrt{\frac{2}{p}s}, \forall s \geq 0, \quad (3.1.13)$$

$$\rho(z) > \frac{1}{\frac{2z}{p} + 1}, \forall z \geq 0. \quad (3.1.14)$$

Démonstration. Pour montrer (3.1.13), soit $\psi_N(t) = s, \forall t \geq 1$, c.-à-d. $t = \sigma(s), t \geq$

1. Par (3.1.11), on a $\frac{p}{2}(t-1)^2 < \psi_N(t)$. Alors $\frac{p}{2}(t-1)^2 < s$ cela implique que

$$t = \sigma(s) < 1 + \sqrt{\frac{2}{p}s}.$$

Par (3.1.12), nous avons :

$$s = \psi_N(t) \leq \frac{p^2+p}{2}(t-1)^2, t \geq 1,$$

cela implique que

$$1 + \sqrt{\frac{2}{p^2+p}s} \leq \sigma(s) = t.$$

Pour montrer (3.1.14), soit $z = -\frac{1}{2}\psi'_N(t), t \in]0, 1]$, c.-à-d. $\rho(z) = t, t \in]0, 1]$, on a $e^{p(\frac{1}{t}-1)} > p\left(\frac{1}{t}-1\right) \geq \left(\frac{1}{t}-1\right)$. Par la définition de $\psi'_N(t)$ dans (3.1.2), nous avons :

$$\begin{aligned}z = -\frac{1}{2}\psi'_N(t) &= -\frac{1}{2}\left(pt - pe^{p(\frac{1}{t}-1)}\right), \\ &> -\frac{1}{2}\left(pt - p\left(\frac{1}{t}-1\right)\right), \\ &\geq -\frac{1}{2}pt + \frac{1}{2}\frac{p}{t} = \frac{p}{2}\left((1-t) + \left(\frac{1}{t}-1\right)\right), \\ &\geq \frac{p}{2}\left(\frac{1}{t}-1\right).\end{aligned}$$

Cela implique que

$$t = \rho(z) > \frac{1}{\left(\frac{2z}{p} + 1\right)}, z \geq 0.$$

Ce qui conclut la preuve du lemme. ■

Soit $\psi_1(t) = pe^{p(\frac{1}{t}-1)}$, $t \in]0, 1]$ et soit $\rho^* :]0, +\infty[\rightarrow]0, 1]$ sa fonction inverse qui est une fonction décroissante. Nous avons alors le lemme suivant :

Lemme 3.1.4. Pour $\psi_1(t)$, on a :

$$\rho^*(z) = \frac{1}{\log\left(\frac{z}{p}\right)^{\frac{1}{p}} + 1}, \forall z > 0, \quad (3.1.15)$$

$$\rho^*(p + 2z) \leq \rho(z), \forall z \geq 0. \quad (3.1.16)$$

Démonstration. Pour montrer (3.1.15), soit $\rho^*(z) = t$, c.-à-d. $z = \psi_1(t) = pe^{p(\frac{1}{t}-1)}$, $p \geq 1$, pour tout $t \in]0, 1]$. Nous avons :

$$e^{p(\frac{1}{t}-1)} = \frac{z}{p},$$

ce qui implique

$$t = \rho^*(z) = \frac{1}{\log\left(\frac{z}{p}\right)^{\frac{1}{p}} + 1}, \forall z > 0.$$

Pour montrer (3.1.16), soit $\rho(z) = t$, c.-à-d. $z = \frac{-1}{2}\psi'_N(t) = \frac{-1}{2}(pt - \psi_1(t))$, pour tout $t \in]0, 1]$. Nous avons :

$$\begin{aligned} z &= \frac{-1}{2}(pt - \psi_1(t)), \\ &\geq \frac{-1}{2}(p - \psi_1(t)), t \in]0, 1], \end{aligned}$$

alors

$$\begin{aligned} p + 2z &\geq \psi_1(t), \\ \rho^*(p + 2z) &\leq t = \rho^*(\psi_1(t)), \\ \rho^*(p + 2z) &\leq \rho(z). \end{aligned}$$

Ce qui termine la preuve du lemme. ■

3.1.2 Analyse de l'algorithme

Lemme 3.1.5. Soient $0 < \theta < 1, v^+ = \frac{v}{\sqrt{1-\theta}}$. Si $\Phi_N(v) \leq \tau$, alors on a :

$$\Phi_N(v^+) < \frac{(p^2 + p)}{2(1 - \theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}\tau} \right)^2.$$

Démonstration. Par le Corollaire 2.4.2, avec $\beta = \frac{1}{\sqrt{1-\theta}} > 1$, et par l'utilisation de (3.1.3) et (3.1.13), on obtient :

$$\begin{aligned} \Phi_N(v^+) &\leq \frac{n}{2} \psi_N''(1) \left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1 \right)^2, \\ &= \frac{n(p^2 + p)}{2} \left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1 \right)^2, \\ &= \frac{n(p^2 + p)}{2(1 - \theta)} \left(\sigma\left(\frac{\tau}{n}\right) - \sqrt{1-\theta} \right)^2, \\ &< \frac{n(p^2 + p)}{2(1 - \theta)} \left(1 + \sqrt{\frac{2}{p}\left(\frac{\tau}{n}\right)} - \sqrt{1-\theta} \right)^2, \\ &\leq \frac{n(p^2 + p)}{2(1 - \theta)} \left(\theta + \sqrt{\frac{2}{p}\left(\frac{\tau}{n}\right)} \right)^2, \quad (1 - \sqrt{1-\theta} \leq \theta), \\ &\leq \frac{(p^2 + p)}{2(1 - \theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}\tau} \right)^2. \end{aligned}$$

Ce qui conclut la preuve du lemme. ■

On désigne par

$$(\Phi_N)_0 = \frac{(p^2 + p)}{2(1 - \theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}\tau} \right)^2; \quad (3.1.17)$$

la borne supérieure de $\Phi_N(v^+)$ au cours du processus de l'algorithme.

Lemme 3.1.6. Soit $\delta(v)$ défini dans (2.2.11). Alors, nous avons :

$$\delta(v) > \sqrt{\frac{p}{2}\Phi_N(v)}. \quad (3.1.18)$$

Démonstration. En utilisant (3.1.11), nous avons :

$$\Phi_N(v) = \sum_{i=1}^n \psi_N(v_i) < \sum_{i=1}^n \frac{1}{2p} \left(\psi_N'(v_i) \right)^2 = \frac{1}{2p} \|\nabla\Phi_N(v)\|^2 = \frac{4}{2p} \delta(v)^2,$$

alors

$$\delta(v) > \sqrt{\frac{p}{2}\Phi_N(v)}.$$

Ce qui termine la preuve du lemme. ■

Remarque 3.1.7. *Tout au long du document, nous supposons que $\tau \geq 1$. En utilisant le Lemme 3.1.6 et l'hypothèse que $\Phi_N(v) \geq \tau$ et $p \geq 1$, nous avons $\delta(v) > \sqrt{\frac{1}{2}}$.*

Lemme 3.1.8. *Soit $\bar{\alpha}$ défini dans le Lemme 2.4.5. Alors, on a :*

$$\bar{\alpha} \geq \frac{1}{p + p(p + 4\delta) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}.$$

Démonstration. En utilisant le Lemme 2.4.6, nous avons $\bar{\alpha} \geq \frac{1}{\psi_N''(\rho(2\delta))}$. D'après (3.1.16) et comme $\frac{1}{\psi_N''(t)}$ est une fonction croissante, on obtient :

$$\frac{1}{\psi_N''(\rho(2\delta))} \geq \frac{1}{\psi_N''(\rho^*(p + 2(2\delta)))}.$$

donc

$$\bar{\alpha} \geq \frac{1}{\psi_N''(\rho^*(p + 4\delta))}.$$

Nous posons $t = \rho^*(p + 4\delta)$ alors on a $t \leq 1$ et

$$\begin{aligned} \psi_N''(t) &= p + \frac{p^2}{t^2} e^{p(\frac{1}{t}-1)}, \\ &= p + \psi_1(t) \left(\frac{p}{t^2}\right). \end{aligned}$$

Nous avons aussi selon (3.1.15) :

$$\begin{aligned} \frac{1}{t^2} &= \frac{1}{(\rho^*(p + 4\delta))^2}, \\ &= \left(1 + \log\left(\frac{p + 4\delta}{p}\right)^{\frac{1}{p}}\right)^2, \end{aligned}$$

et

$$\psi_1(t) = \psi_1(\rho^*(p + 4\delta)) = p + 4\delta.$$

Donc, on a :

$$\begin{aligned}\psi''_N(\rho^*(p+4\delta)) &= p + \psi_1(t) \left(\frac{p}{t^2}\right), \\ &= p + p(p+4\delta) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2.\end{aligned}$$

Enfin, nous obtenons :

$$\begin{aligned}\bar{\alpha} &\geq \frac{1}{\psi''_N(\rho^*(p+4\delta))}, \\ &= \frac{1}{p + p(p+4\delta) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}.\end{aligned}$$

Ce qui conclut la preuve du lemme. ■

Nous posons :

$$\tilde{\alpha} = \frac{1}{p + p(p+4\delta) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}. \quad (3.1.19)$$

$\tilde{\alpha}$ est le pas du déplacement tel que $\tilde{\alpha} \leq \bar{\alpha}$.

Lemme 3.1.9. Soit $\tilde{\alpha}$ le pas de déplacement défini dans (3.1.19). Si en prenant $\Phi_N(v) \geq 1$ alors on a :

$$f(\tilde{\alpha}) \leq \frac{-\sqrt{\frac{p}{2}\Phi_N}}{(\sqrt{2p} + p(\sqrt{2p} + 4)) \left(1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}(\Phi_N)_0}}{p}\right)^{\frac{1}{p}}\right)^2}. \quad (3.1.20)$$

Démonstration. En utilisant le Lemme 2.4.8 avec $\alpha = \tilde{\alpha}$ et (3.1.19). On obtient :

$$\begin{aligned}f(\tilde{\alpha}) &\leq -\tilde{\alpha}\delta^2, \\ &= \frac{-\delta^2}{p + p(p+4\delta) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}, \\ &< \frac{-\delta^2}{p + p(\sqrt{2p} + 4)\delta \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}, \delta > \frac{1}{\sqrt{2}},\end{aligned}$$

puisque $\log\left(1 + \frac{4\delta}{p}\right) \geq 0$ et $\delta \geq \sqrt{\frac{p}{2}}$, nous avons :

$$f(\tilde{\alpha}) \leq \frac{-\delta}{\left(p\sqrt{\frac{2}{p}} + p(\sqrt{2p} + 4)\right) \left(1 + \log\left(\frac{p+4\delta}{p}\right)^{\frac{1}{p}}\right)^2}.$$

Soit $g_1(x) = \frac{x}{\left(1 + \log\left(\frac{p+4x}{p}\right)^{\frac{1}{p}}\right)^2}, \forall x \in \mathfrak{R}_{++}$ une fonction croissante et nous avons $\delta \geq \sqrt{\frac{p}{2}}\Phi_N$ alors

$$f(\tilde{\alpha}) \leq \frac{-\sqrt{\frac{p}{2}}\Phi_N}{\left(\sqrt{2p} + p(\sqrt{2p} + 4)\right) \left(1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}}\Phi_N}{p}\right)^{\frac{1}{p}}\right)^2},$$

Soit $g_2(x) = \frac{1}{\left(1 + \log\left(\frac{p+4x}{p}\right)^{\frac{1}{p}}\right)^2}, \forall x \in \mathfrak{R}_{++}$ une fonction décroissante et $\Phi_N \leq (\Phi_N)_0$, on obtient

$$f(\tilde{\alpha}) \leq \frac{-\sqrt{\frac{p}{2}}\Phi_N}{\left(\sqrt{2p} + p(\sqrt{2p} + 4)\right) \left(1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}}(\Phi_N)_0}{p}\right)^{\frac{1}{p}}\right)^2}.$$

Cela complète la preuve. ■

Après la mise à jour de μ à $(1 - \theta)\mu$, nous avons :

$$\Phi_N(v^+) \leq (\Phi_N)_0 = \frac{(p^2 + p)}{2(1 - \theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}}\tau \right)^2.$$

La décroissance de Φ_N dans chaque itération interne est donnée par (3.1.20), nous avons :

$$(\Phi_N)_{k+1} \leq (\Phi_N)_k - \bar{\kappa}(\Phi_N)^{1-\gamma}, \quad k = 0, 1, \dots, K - 1,$$

avec

$$\gamma = \frac{1}{2}, \bar{\kappa} = \frac{\sqrt{p}}{\sqrt{2} \left(\sqrt{2p} + p(\sqrt{2p} + 4) \right) \left(1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}}(\Phi_N)_0}{p}\right)^{\frac{1}{p}} \right)^2}.$$

Donc, par le Lemme 2.5.1 pour $t_k = (\Phi_N)_k$, alors le nombre d'itérations internes dans l'itération externe K se représente comme suit :

$$K \leq \frac{2\sqrt{2} \left(\sqrt{2p} + p(\sqrt{2p} + 4) \right)}{\sqrt{p}} \left(1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}}(\Phi_N)_0}{p}\right)^{\frac{1}{p}} \right)^2 (\Phi_N)_0^{1/2}.$$

D'après (2.5.1), le nombre total d'itérations pour trouver une solution primale-duale optimale est majoré par

$$\frac{2\sqrt{2}(\sqrt{2p} + p(\sqrt{2p} + 4))}{\sqrt{p}} \left(1 + \log \left(\frac{p + 4\sqrt{\frac{p}{2}(\Phi_N)_0}}{p} \right)^{\frac{1}{p}} \right)^2 (\Phi_N)_0^{1/2} \frac{\log \frac{n}{\varepsilon}}{\theta}. \quad (3.1.21)$$

◆ Pour les méthodes à long-pas avec $\tau = O(n)$ et $\theta = \Theta(1)$, on a :

$$O \left(\sqrt{p^5 n} (\log pn)^2 \log \frac{n}{\varepsilon} \right),$$

itérations, où $(\Phi_N)_0 = O(p^2 n)$.

◆ Pour les méthodes à court-pas avec $\tau = O(1)$ et $\theta = \Theta(\frac{1}{\sqrt{n}})$, on a :

$$O \left(\sqrt{p^5 n} \log \frac{n}{\varepsilon} \right),$$

itérations, où $(\Phi_N)_0 = O(p^2)$.

◆ Si $p = O(1)$, nous obtenons $O(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon})$ itérations pour les méthodes à long-pas et $O(\sqrt{n} \log \frac{n}{\varepsilon})$ itérations pour les méthodes à court-pas qui est la meilleure borne de complexité connue j'ausqu'à présent.

3.1.3 Tests numériques

Pour prouver l'efficacité de notre nouvelle fonction noyau ψ_N , on va effectuer des tests numériques comparatifs avec celle donnée par Bai et al. [7] (cas de $p = 1$).

L'**Algorithme 5** a été codé en MATLAB 2009. Nous donnons ici quelques clarifications nécessaires dans l'implémentation :

- Nous utilisons à chaque itération interne :

1. Un pas de déplacement théorique $\alpha_{théor} = \tilde{\alpha} = \frac{1}{p + p(p+4\delta) \left(1 + \log \left(\frac{p+4\delta}{p} \right)^{\frac{1}{p}} \right)^2}$.
2. Un pas de déplacement pratique α_{prat} telle que

$$x^+ = x + \alpha_{prat} \Delta x > 0 \text{ et } z^+ = z + \alpha_{prat} \Delta z > 0.$$

où $\alpha_{prat} = \rho \min(\alpha_x, \alpha_z), \rho \in]0, 1[$, avec

$$\alpha_x = \min \left\{ 1, \min_i \left\{ -\frac{x_i}{\Delta x_i}, \Delta x_i < 0 \right\} \right\}, \alpha_z = \min \left\{ 1, \min_i \left\{ -\frac{z_i}{\Delta z_i}, \Delta z_i < 0 \right\} \right\}.$$

- Dans les tests, on prend :

$$\varepsilon = 10^{-4}, \theta = \{0.01, 0.5, 0.9, 0.99\}, \rho = 0.99 \text{ et } \tau = 1.$$

- Nous indiquons $I_{t\text{ext}}$, $I_{t\text{rint}}$ et $T(s)$ le nombre d'itérations externes, internes et le temps de calcul par seconde respectivement, pour obtenir la solution optimale.

- Nous rappelons que notre première nouvelle fonction noyau est définie par :

$$\psi_N = \frac{p(t^2 - 1)}{2} - \int_1^t p e^{p(\frac{1}{x} - 1)} dx, p \geq 1,$$

on choisit $p = \{1, 1.1, 2, 5\}$.

Nous considérons le problème (PQC) sous la forme suivante :

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ x \geq 0. \end{cases}$$

Exemple 1

$$A = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, c = \begin{pmatrix} -2 \\ -4 \\ 0 \end{pmatrix}, Q = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Le point initial strictement réalisable est :

$$x^0 = \begin{pmatrix} 0.3262 & 1.3261 & 0.3477 \end{pmatrix}^t, y^0 = \begin{pmatrix} 0 & -2.0721 \end{pmatrix}^t, \\ z^0 = \begin{pmatrix} 0.7247 & 0.7247 & 2.0722 \end{pmatrix}^t.$$

La solution optimale primale-duale obtenue est :

$$x^* = \begin{pmatrix} 0.4999 & 1.4998 & 0.0003 \end{pmatrix}^t, y^* = \begin{pmatrix} 0 & -1 \end{pmatrix}^t, \\ z^* = \begin{pmatrix} 0.0001 & 0 & 0.9999 \end{pmatrix}^t.$$

Exemple 2

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 5 \end{pmatrix}, c = \begin{pmatrix} -4 \\ -6 \\ 0 \\ 0 \end{pmatrix}, Q = \begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Le point initial strictement réalisable est :

$$x^0 = \begin{pmatrix} 0.9683 & 0.5775 & 0.4543 & 1.1444 \end{pmatrix}^t, y^0 = \begin{pmatrix} -0.9184 & -1.1244 \end{pmatrix}^t,$$

$$z^0 = \begin{pmatrix} 0.7612 & 0.9141 & 0.9185 & 1.1244 \end{pmatrix}^t.$$

La solution optimale primale-duale obtenue est :

$$x^* = \begin{pmatrix} 1.1288 & 0.7742 & 0.0971 & 0.0003 \end{pmatrix}^t, y^* = \begin{pmatrix} -0.0017 & -1.0318 \end{pmatrix}^t,$$

$$z^* = \begin{pmatrix} 0 & 0 & 0.0003 & 1.0321 \end{pmatrix}^t.$$

Exemple 3

$$A = \begin{pmatrix} 1 & 1.2 & 1 & 1.8 & 0 \\ 3 & -1 & 1.5 & -2 & 1 \\ -1 & 2 & -3 & 4 & 2 \end{pmatrix}, b = \begin{pmatrix} 9.31 \\ 5.45 \\ 6.60 \end{pmatrix}, c = \begin{pmatrix} 1 \\ -1.5 \\ 2 \\ 1.5 \\ 3 \end{pmatrix},$$

$$Q = \begin{pmatrix} 20 & 1.2 & 0.5 & 0.5 & -1 \\ 1.2 & 32 & 1 & 1 & 1 \\ 0.5 & 1 & 14 & 1 & 1 \\ 0.5 & 1 & 1 & 15 & 1 \\ -1 & 1 & 1 & 1 & 16 \end{pmatrix}.$$

Le point initial strictement réalisable est :

$$x^0 = \begin{pmatrix} 2.4539 & 0.7875 & 1.5838 & 2.4038 & 1.3074 \end{pmatrix}^t,$$

$$y^0 = \begin{pmatrix} 20.5435 & 9.4781 & 4.3927 \end{pmatrix}^t,$$

$$z^0 = \begin{pmatrix} 7.1215 & 7.9763 & 8.3150 & 6.8686 & 7.9750 \end{pmatrix}^t.$$

La solution optimale primale-duale obtenue est :

$$x^* = \begin{pmatrix} 2.6321 & 0.7019 & 1.3994 & 2.4643 & 1.0846 \end{pmatrix}^t,$$

$$y^* = \begin{pmatrix} 25.2686 & 11.7725 & 5.2567 \end{pmatrix}^t,$$

$$z^* = \begin{pmatrix} 0.0757 & 0.2839 & 0.1424 & 0.0808 & 0.1837 \end{pmatrix}^t * 10^{-4}.$$

Exemple 4

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix}, b = \begin{pmatrix} 11.651 \\ 16.672 \\ 21.295 \end{pmatrix},$$

$$c = \begin{pmatrix} -0.5 \\ -1 \\ 0 \\ 0 \\ -0.5 \\ 0 \\ 0 \\ -1 \\ -0.5 \\ -1 \end{pmatrix}, Q = \begin{pmatrix} 30 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 21 & 0 & 1 & -1 & 1 & 0 & 1 & 0.5 & 1 & \\ 1 & 0 & 15 & -0.5 & -2 & 1 & 0 & 1 & 1 & 1 & \\ 1 & 1 & -0.5 & 30 & 3 & -1 & 1 & -1 & 0.5 & 1 & \\ 1 & -1 & -2 & 3 & 27 & 1 & 0.5 & 1 & 1 & 1 & \\ 1 & 1 & 1 & -1 & 1 & 16 & -0.5 & 0.5 & 0 & 1 & \\ 1 & 0 & 0 & 1 & 0.5 & -0.5 & 8 & 1 & 1 & 1 & \\ 1 & 1 & 1 & -1 & 1 & 0.5 & 1 & 24 & 1 & 1 & \\ 1 & 0.5 & 1 & 0.5 & 1 & 0 & 1 & 1 & 39 & 1 & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 11 \end{pmatrix}.$$

Le point initial strictement réalisable est :

$$x^0 = \begin{pmatrix} 0.9491 & 0.6121 & 1.8477 & 1.8115 & 1.2511 & 2.5211 & 1.5062 & 1.5658 & 0.8207 & 1.1289 \end{pmatrix}^t,$$

$$y^0 = \begin{pmatrix} 4.3800 & 19.9367 & 4.5679 \end{pmatrix}^t,$$

$$z^0 = \begin{pmatrix} 3.8902 & 4.4625 & 3.9788 & 3.6606 & 3.9018 & 3.5561 & 3.8760 & 3.7190 & 3.9138 & 4.3396 \end{pmatrix}^t.$$

La solution optimale primale-duale obtenue est :

$$x^* = \begin{pmatrix} 0.9639 & 0.5096 & 1.7399 & 1.9049 & 1.2435 & 2.6266 & 1.3229 & 1.6170 & 0.8240 & 0.8976 \end{pmatrix}^t,$$

$$y^* = \begin{pmatrix} 4.2429 & 22.3605 & 5.1916 \end{pmatrix}^t,$$

$$z^* = \begin{pmatrix} 0.1035 & 0.1958 & 0.0574 & 0.0524 & 0.0803 & 0.0380 & 0.0754 & 0.0617 & 0.1211 & 0.1112 \end{pmatrix}^t * 10^{-4}.$$

Exemple 5 : Exemple de taille variable

$$n = 2m, A(i, j) = \begin{cases} 0 & \text{si } i \neq j \text{ et } j \neq i + m, \\ 1 & \text{si } i = j \text{ et } j = i + m, \end{cases}, Q(i, j) = \begin{cases} 0 & \text{si } i \neq j, \\ 1 & \text{si } i = j, \end{cases}$$

$$c(i) = -1, c(i + m) = 0, b(i) = 2, \text{ pour } i = 1, \dots, m.$$

Le point initial strictement réalisable est :

$$x^0(i) = x^0(i + m) = 1, y^0(i) = -1, z^0(i) = 1, z^0(i + m) = 2 \text{ pour } i = 1, \dots, m.$$

La solution optimale primale-duale obtenue est :

$$x^*(i) = \frac{3}{2}, x^*(i + m) = \frac{1}{2}, y^*(i) = \frac{1}{2}, z^*(i) = 0, z^*(i + m) = 0, \text{ pour } i = 1, \dots, m.$$

Nous résumons dans les tableaux de 3 à 6 les résultats numériques des exemples de 1 à 4. Et les résultats numériques de l'exemple 5 sont présentés dans les tableaux de 7 à 10.

p		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$p = 1$	<i>Itrint</i>	459	525	735	938
	<i>Itrext</i>	1080	1111	1408	1372
	$T(s)$	10.75	14.88	24.22	61.63
$p = 1.1$	<i>Itrint</i>	436	500	701	898
	<i>Itrext</i>	1074	1101	1404	1368
	$T(s)$	10.43	14.54	23.36	53.12
$p = 2$	<i>Itrint</i>	371	430	607	799
	<i>Itrext</i>	1043	1081	1379	1348
	$T(s)$	9.56	13.28	22.04	58.49
$p = 5$	<i>Itrint</i>	471	546	772	1040
	<i>Itrext</i>	1013	1051	1354	1330
	$T(s)$	10.31	14.35	23.78	52.97

Tableau 3 : Résultats numériques des exemples de 1 à 4 pour différentes valeurs de p avec $\theta = 0.01$ et $\alpha = \alpha_{théor}$.

p		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$p = 1$	<i>Itrint</i>	630	756	1042	1349
	<i>Itrext</i>	16	17	21	20
	$T(s)$	5.94	8.82	17.31	41.17
$p = 1.1$	<i>Itrint</i>	598	719	944	1286
	<i>Itrext</i>	16	17	21	20
	$T(s)$	5.72	8.58	14.12	33.40
$p = 2$	<i>Itrint</i>	498	563	830	1097
	<i>Itrext</i>	16	16	21	20
	$T(s)$	4.55	6.61	11.61	36.35
$p = 5$	<i>Itrint</i>	547	661	923	1289
	<i>Itrext</i>	15	16	20	20
	$T(s)$	5.72	8.61	16.11	38.19

Tableau 4 : Résultats numériques des exemples de 1 à 4 pour différentes valeurs de p avec $\theta = 0.5$ et $\alpha = \alpha_{théor.}$

p		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$p = 1$	<i>Itrint</i>	4834	5552	8684	9003
	<i>Itrext</i>	3	3	4	3
	$T(s)$	86.78	129.44	267.60	548.65
$p = 1.1$	<i>Itrint</i>	4472	5161	8058	8415
	<i>Itrext</i>	3	3	4	3
	$T(s)$	77.76	120.39	238.56	476.11
$p = 2$	<i>Itrint</i>	3136	3625	5632	6044
	<i>Itrext</i>	3	3	4	3
	$T(s)$	60.05	94.02	211.70	469.32
$p = 5$	<i>Itrint</i>	2778	3235	3658	5371
	<i>Itrext</i>	3	3	3	3
	$T(s)$	71.16	113.13	162.84	481.52

Tableau 5 : Résultats numériques des exemples de 1 à 4 pour différentes valeurs de p avec $\theta = 0.99$ et $\alpha = \alpha_{théor.}$

	$\theta = 0.01$			$\theta = 0.5$			$\theta = 0.99$		
Exp	<i>Itrint</i>	<i>Itrext</i>	$T(s)$	<i>Itrint</i>	<i>Itrext</i>	$T(s)$	<i>Itrint</i>	<i>Itrext</i>	$T(s)$
1	12	1038	6.98	9	16	0.22	7	3	0.13
2	14	1076	9.54	11	16	0.28	7	3	0.17
3	19	1355	16.82	15	20	0.47	6	3	0.18
4	25	1343	33.09	20	20	0.98	6	3	0.43

Tableau 6 : Résultats numériques des exemples de 1 à 4 pour $p = 1$ et différentes valeurs de θ avec $\alpha = \alpha_{prat}$.

p	(m, n)	(10, 20)	(30, 60)	(50, 100)	(100, 200)
$p = 1$	<i>Itrint</i>	1187	2100	2765	4038
	<i>Itrext</i>	1297	1389	1434	1498
	$T(s)$	121.08	536.66	1102.18	3485.54
$p = 1.1$	<i>Itrint</i>	1142	2032	2677	3916
	<i>Itrext</i>	1294	1388	1433	1497
	$T(s)$	117.66	531.53	1082.76	3420.59
$p = 2$	<i>Itrint</i>	1041	1890	2507	3687
	<i>Itrext</i>	1280	1379	1427	1492
	$T(s)$	103.40	498.06	1013.75	3215.86
$p = 5$	<i>Itrint</i>	1372	2519	3348	4930
	<i>Itrext</i>	1266	1371	1420	1488
	$T(s)$	127.79	718.37	1243.88	4046.03

Tableau 7 : Résultats numériques de l'exemple 5 pour différentes valeurs de p avec $\theta = 0.01$ et $\alpha = \alpha_{théor}$.

p	(m, n)	(10, 20)	(30, 60)	(50, 100)	(100, 200)
$p = 1$	<i>Itrint</i>	1799	3639	4910	7866
	<i>Itrext</i>	19	21	21	22
	$T(s)$	95.36	766.68	1455.22	5463.59
$p = 1.1$	<i>Itrint</i>	1722	3499	4723	7565
	<i>Itrext</i>	19	21	21	22
	$T(s)$	90.17	601.35	1368.03	5647.23
$p = 2$	<i>Itrint</i>	1493	2906	4120	6560
	<i>Itrext</i>	19	20	21	22
	$T(s)$	78.04	497.92	1163.21	4540.73
$p = 5$	<i>Itrint</i>	1745	3328	4648	7226
	<i>Itrext</i>	19	20	21	22
	$T(s)$	101.06	704.12	1492.60	5566.29

Tableau 8 : Résultats numériques de l'exemple 5 pour différentes valeurs de p avec $\theta = 0.5$ et $\alpha = \alpha_{théor.}$

p	(m, n)	(10, 20)	(30, 60)	(50, 100)	(100, 200)
$p = 1$	<i>Itrint</i>	4282	9697	13331	20702
	<i>Itrext</i>	6	7	7	7
	$T(s)$	322.53	2981.57	5203.76	18021.87
$p = 1.1$	<i>Itrint</i>	4063	9169	12650	19633
	<i>Itrext</i>	6	7	7	7
	$T(s)$	293.87	2648.62	5947.69	21314.96
$p = 2$	<i>Itrint</i>	3206	7249	9945	15329
	<i>Itrext</i>	6	7	7	7
	$T(s)$	258.12	1808.19	4147.40	14829.11
$p = 5$	<i>Itrint</i>	3231	6039	9481	14234
	<i>Itrext</i>	6	6	7	7
	$T(s)$	346.88	2012.71	5229.10	17515.58

Tableau 9 : Résultats numériques de l'exemple 5 pour différentes valeurs de p avec $\theta = 0.9$ et $\alpha = \alpha_{théor.}$

θ	(m, n)	(10, 20)	(30, 60)	(50, 100)	(100, 200)	(250, 500)	(500, 1000)
$\theta = 0.01$	<i>Itrint</i>	32	58	76	108	177	237
	<i>Itrext</i>	1259	1366	1423	1495	1581	1648
	$T(s)$	56.00	194.44	338.83	746.10	3229.66	18568.48
$\theta = 0.5$	<i>Itrint</i>	19	20	22	23	26	48
	<i>Itrext</i>	19	20	21	22	23	24
	$T(s)$	1.96	5.86	10.42	24.41	232.28	3072.50
$\theta = 0.9$	<i>Itrint</i>	13	13	15	15	16	18
	<i>Itrext</i>	6	6	7	7	7	8
	$T(s)$	1.42	4.51	7.19	16.26	143.22	1156.67

Tableau 10 : Résultats numériques de l'exemple 5 pour différentes valeurs de θ et $p = 1$ avec $\alpha = \alpha_{prat}$.

Commentaires

D'après les résultats numériques obtenus dans les tableaux de 3 à 5 et de 7 à 9, on constate que :

- Le nombre d'itérations et le temps de calcul pour obtenir la solution optimale des exemples testés dépendent de valeurs du paramètre p et le paramètre barrière θ . En fait, $p = 2$ ou $p = 5$ offre des meilleurs résultats que $p = 1$, c'est le cas de la fonction noyau de Bai.

- Le nombre d'itérations externes est considérablement réduit lorsque $\theta = 0.99$ pour les exemples de 1 à 4 et $\theta = 0.9$ pour l'exemple 5, par contre $\theta = 0.5$ offre le plus petit nombre d'itérations total pour tous les exemples testés.

En comparant les résultats des tableaux 3 et 7 par les résultats des tableaux 6 et 10 respectivement, on remarque que :

- Le nombre d'itérations totale et le temps de calculs pour $p = 1$ est considérablement diminue lorsque $\alpha = \alpha_{prat}$ en le comparant par $\alpha_{théor}$, malheureusement pour $p = 1.1, p = 2$ et $p = 5$, l'algorithme ne converge pas.

Cela confirme nos résultats théoriques et consolide notre objectif d'améliorer le comportement numérique de l'algorithme.

La section suivante est consacrée à notre deuxième nouvelle fonction noyau.

Cette fonction noyau est paramétrée avec un terme barrière polynomial (de Laurent). Les propriétés requises dans l'analyse de l'algorithme obtenu sont présentées le long de ce chapitre. En fait, le contenu de cette partie a fait l'objet de notre deuxième article [19].

3.2 Nouvelle fonction noyau avec terme barrière polynomial (de Laurent)

Nous définissons notre deuxième nouvelle fonction noyau $\psi_B(t)$ comme suit :

$$\psi_B(t) = t^2 - t + \frac{t^{-(p+1)} - 1}{p+1}, \quad p > 0. \quad (3.2.1)$$

Nous donnons les trois premières dérivées par rapport à t comme suit :

$$\psi'_B(t) = 2t - 1 - t^{-(p+2)}, \quad (3.2.2)$$

$$\psi''_B(t) = 2 + (p+2)t^{-(p+3)}, \quad (3.2.3)$$

$$\psi'''_B(t) = -(p+2)(p+3)t^{-(p+4)}. \quad (3.2.4)$$

Il est évident que nous avons :

$$\psi''_B(t) > 2 > 0, \quad \forall t > 0, \quad (3.2.5)$$

$$\psi_B(1) = \psi'_B(1) = 0. \quad (3.2.6)$$

3.2.1 Eligibilité de la nouvelle fonction noyau

Le lemme suivant sert à prouver certaines propriétés d'éligibilité de notre nouvelle fonction noyau $\psi_B(t)$ qui est définie dans (3.2.1).

Lemme 3.2.1. *Soit $\psi_B(t)$ définie dans (3.2.1). Alors, on a :*

$$\psi'''_B(t) < 0, \quad \forall t > 0, \quad (3.2.7)$$

$$t\psi_B''(t) - \psi_B'(t) > 0, \forall t > 0, \quad (3.2.8)$$

$$t\psi_B''(t) + \psi_B'(t) > 0, \forall t > 0, \quad (3.2.9)$$

$$\psi_B''(t) \psi_B'(\beta t) - \beta \psi_B'(t) \psi_B''(\beta t) > 0, \forall \beta > 1, \forall t > 1. \quad (3.2.10)$$

Démonstration. Pour montrer (3.2.7), en utilisant (3.2.4), on obtient :

$$\psi_B'''(t) < 0, \forall t > 0.$$

Pour montrer (3.2.8), on utilise (3.2.2), (3.2.3) et la positivité de t et p , on obtient :

$$t\psi_B''(t) - \psi_B'(t) = 1 + (p+3)t^{-(p+2)} > 0.$$

Maintenant, nous montrons que (3.2.9) est valable pour tout $t > 0$. À ce propos, soit $0 < t \leq 1$, on a :

$$\begin{aligned} t\psi_B''(t) + \psi_B'(t) &= 4t + (p+1)t^{-(p+2)} - 1, \\ &> p > 0. \end{aligned}$$

D'autre part, pour $t > 1$, on a :

$$\begin{aligned} t\psi_B''(t) + \psi_B'(t) &= 4t + (p+1)t^{-(p+2)} - 1, \\ &> 3 + (p+1)t^{-(p+2)} > 0. \end{aligned}$$

Pour montrer (3.2.10), on considère $f(\beta)$ comme suit :

$$f(\beta) = \psi_B''(t) \psi_B'(\beta t) - \beta \psi_B'(t) \psi_B''(\beta t), \forall \beta > 1, \forall t > 1.$$

on a :

$$f'(\beta) = \psi_B''(\beta t) (t\psi_B''(t) - \psi_B'(t)) - \beta t \psi_B'(t) \psi_B'''(\beta t), \forall \beta > 1,$$

comme ψ_B vérifie (3.2.7), (3.2.8) et par (b) de Lemme 2.1.6, on obtient

$$f'(\beta) > 0, \forall \beta > 1 \text{ et } f(1) = 0,$$

alors

$$f(\beta) > 0, \forall \beta > 1.$$

Ce qui termine la preuve du lemme. ■

En préparation pour la suite, nous présentons quelques résultats techniques de notre nouvelle fonction noyau (3.2.1).

Lemme 3.2.2. Pour $\psi_B(t)$, on a :

$$(t-1)^2 \leq \psi_B(t) \leq \frac{1}{4} \left(\psi'_B(t) \right)^2, \quad \forall t > 0. \quad (3.2.11)$$

$$\psi_B(t) \leq \frac{1}{2} \psi''_B(1) (t-1)^2 = \frac{p+4}{2} (t-1)^2, \quad \forall t \geq 1. \quad (3.2.12)$$

Démonstration. Pour montrer (3.2.11), en utilisant (3.2.5), on a :

$$\psi_B(t) = \int_1^t \int_1^x \psi''_B(y) dy dx \geq \int_1^t \int_1^x 2 dy dx = (t-1)^2,$$

et

$$\begin{aligned} \psi_B(t) &= \int_1^t \int_1^x \psi''_B(y) dy dx, \\ &\leq \frac{1}{2} \int_1^t \int_1^x 2 \psi''_B(y) dy dx, \\ &\leq \frac{1}{2} \int_1^t \int_1^x \psi''_B(x) \psi''_B(y) dy dx, \\ &\leq \frac{1}{2} \int_1^t \psi''_B(x) \psi'_B(x) dx = \frac{1}{4} \left(\psi'_B(t) \right)^2. \end{aligned}$$

Pour montrer (3.2.12), en utilisant (3.2.6), (3.2.7) et le développement de Taylor, nous avons pour certain ζ ($1 \leq \zeta \leq t$) :

$$\begin{aligned} \psi_B(t) &= \psi_B(1) + \psi'_B(1) (t-1) + \frac{1}{2} \psi''_B(1) (t-1)^2 + \frac{1}{6} \psi'''_B(\zeta) (\zeta-1)^3, \\ &\leq \frac{1}{2} \psi''_B(1) (t-1)^2 = \frac{p+4}{2} (t-1)^2. \end{aligned}$$

Ce qui conclut la preuve du lemme. ■

Lemme 3.2.3. Soient $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de $\psi_B(t)$ et $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de $-\frac{1}{2}\psi'_B(t)$, on a :

$$1 + \sqrt{\frac{2}{p+4}}s \leq \sigma(s) \leq 1 + \sqrt{s}, \forall s \geq 0, \quad (3.2.13)$$

$$\rho(z) > \frac{1}{(2z+2)^{\frac{1}{p+2}}}, \forall z \geq 0. \quad (3.2.14)$$

Démonstration. Pour montrer (3.2.13), soit $\psi_B(t) = s, \forall t \geq 1$, c.-à-d., $t = \sigma(s)$. Par (3.2.11), on a $(t-1)^2 \leq \psi_B(t) = s$ cela implique que

$$t = \sigma(s) \leq 1 + \sqrt{s}.$$

Par (3.2.12), on a $s = \psi_B(t) \leq \frac{p+4}{2}(t-1)^2, \forall t \geq 1$, cela implique

$$1 + \sqrt{\frac{2}{p+4}}s \leq \sigma(s) = t.$$

Pour montrer (3.2.14), soit $z = -\frac{1}{2}\psi'_B(t), t \in]0, 1]$, c.-à-d., $\rho(z) = t$. Par (3.2.2), on a :

$$z = \frac{-1}{2} \left(2t - 1 - t^{-(p+2)} \right) > \frac{1}{2}(-2 + t^{-(p+2)}),$$

cela implique que

$$t = \rho(z) > \frac{1}{(2z+2)^{\frac{1}{p+2}}}, z \geq 0.$$

Ce qui termine la preuve du lemme ■

Lemme 3.2.4. Soient $0 < \theta < 1, v^+ = \frac{v}{\sqrt{1-\theta}}$. Si $\Phi_B(v) \leq \tau$, alors on a :

$$\Phi_B(v^+) \leq \frac{(\tau + 2\sqrt{n\tau} + \theta n)}{(1-\theta)}. \quad (3.2.15)$$

Démonstration. Par le Théorème 2.4.1 avec $\beta = \frac{1}{\sqrt{1-\theta}} > 1$, en utilisant (3.2.13)

et $\psi_B(t) \leq t^2 - 1, \forall t \geq 1$, on obtient :

$$\begin{aligned}
 \Phi_B(v^+) &\leq n\psi_B\left(\frac{\sigma\left(\frac{\Phi_B(v)}{n}\right)}{\sqrt{1-\theta}}\right), \\
 &\leq n\psi_B\left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right), \\
 &\leq n\left(\left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right)^2 - 1\right), \\
 &= \frac{n}{(1-\theta)}\left(\left(\sigma\left(\frac{\tau}{n}\right)\right)^2 - (1-\theta)\right), \\
 &\leq \frac{n}{(1-\theta)}\left(\left(1 + \sqrt{\frac{\tau}{n}}\right)^2 - (1-\theta)\right), \\
 &\leq \frac{n}{(1-\theta)}\left(\frac{\tau}{n} + 2\sqrt{\frac{\tau}{n}} + \theta\right), \\
 &= \frac{(\tau + 2\sqrt{n\tau} + \theta n)}{(1-\theta)}.
 \end{aligned}$$

Ce qui conclut la preuve du lemme. ■

Par le Corollaire 2.4.2, avec $\beta = \frac{1}{\sqrt{1-\theta}} > 1$ et (3.2.13), on obtient :

$$\begin{aligned}
 \Phi_B(v^+) &\leq \frac{n}{2}\psi_B''(1)\left(\frac{\sigma\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2, \\
 &= \frac{n(p+4)}{2(1-\theta)}\left(\sigma\left(\frac{\tau}{n}\right) - \sqrt{1-\theta}\right)^2, \\
 &\leq \frac{n(p+4)}{2(1-\theta)}\left(1 + \sqrt{\frac{\tau}{n}} - \sqrt{1-\theta}\right)^2, \\
 &\leq \frac{n(p+4)}{2(1-\theta)}\left(\theta + \sqrt{\frac{\tau}{n}}\right)^2, \quad (1 - \sqrt{1-\theta} \leq \theta, \quad 0 < \theta < 1), \\
 &= \frac{(p+4)}{2(1-\theta)}(\theta\sqrt{n} + \sqrt{\tau})^2.
 \end{aligned}$$

Donc

$$\Phi_B(v^+) \leq \frac{(p+4)}{2(1-\theta)}(\theta\sqrt{n} + \sqrt{\tau})^2. \quad (3.2.16)$$

On note par $(\Phi_B)_0$; la borne supérieure de $\Phi_B(v^+)$ au cours du processus de l'algorithme qui est définie par l'une des deux formules (3.2.15) et (3.2.16).

Lemme 3.2.5. Soit $\delta(v)$ défini dans (2.2.11). Alors, nous avons :

$$\delta(v) \geq \sqrt{\Phi_B(v)}. \quad (3.2.17)$$

Démonstration. En utilisant (3.2.11), on a :

$$\Phi_B(v) = \sum_{i=1}^n \psi_B(v_i) \leq \sum_{i=1}^n \frac{1}{4} \left(\psi'_B(v_i) \right)^2 = \frac{1}{4} \|\nabla \Phi_B(v)\|^2 = \delta(v)^2,$$

alors $\delta(v) \geq \sqrt{\Phi_B(v)}$.

Ce qui termine la preuve du lemme. ■

Remarque 3.2.6. Dans la suite, nous supposons que $\tau \geq 1$. En utilisant le Lemme 3.2.5 et l'hypothèse $\Phi_B(v) \geq \tau$, on a $\delta(v) \geq 1$.

Lemme 3.2.7. Soit $\bar{\alpha}$ celui défini dans Lemme 2.4.5, nous avons :

$$\bar{\alpha} \geq \frac{1}{2 + (p+2)(4\delta+2)^{\frac{p+3}{p+2}}}.$$

Démonstration. En utilisant le Lemme 2.4.6 et (3.2.3), alors on a :

$$\bar{\alpha} \geq \frac{1}{\psi''(\rho(2\delta))} = \frac{1}{2 + (p+2)(\rho(2\delta))^{-(p+3)}},$$

Par (3.2.14), on trouve :

$$\bar{\alpha} > \frac{1}{2 + (p+2)(4\delta+2)^{\frac{p+3}{p+2}}}.$$

Ce qui conclut la preuve du lemme. ■

Soit

$$\bar{\tilde{\alpha}} = \frac{1}{2 + (p+2)(4\delta+2)^{\frac{p+3}{p+2}}}. \quad (3.2.18)$$

$\bar{\tilde{\alpha}}$ est le pas de déplacement tel que $\bar{\tilde{\alpha}} \leq \bar{\alpha}$.

Lemme 3.2.8. Pour $\bar{\tilde{\alpha}}$ le pas de déplacement défini dans (3.2.18) et par l'hypothèse $\Phi_B(v) \geq 1$. Alors, on a :

$$f(\bar{\tilde{\alpha}}) < -\frac{(\Phi_B)^{\frac{p+1}{2(p+2)}}}{36(p+4)}. \quad (3.2.19)$$

Démonstration. En utilisant le Lemme 2.4.8 avec $\alpha = \bar{\alpha}$ et (3.2.17), on obtient :

$$\begin{aligned} f(\bar{\alpha}) &\leq \frac{-\delta^2}{2 + (p+2)(4\delta+2)^{\frac{p+3}{p+2}}}, \\ &< \frac{-\delta^2}{(p+4)(4\delta+2)^{\frac{p+3}{p+2}}}, \\ &\leq \frac{-\delta^2}{(p+4)(6\delta)^{\frac{p+3}{p+2}}}, \delta \geq 1, \\ &\leq -\frac{(\Phi_B)^{\frac{p+1}{2(p+2)}}}{36(p+4)}. \end{aligned}$$

Ce qui termine la preuve du lemme. ■

3.2.2 Complexité de l'algorithme

La décroissance de la fonction barrière Φ_B dans chaque itération interne est donnée par (3.2.19), nous avons :

$$(\Phi_B)_{k+1} \leq (\Phi_B)_k - \bar{\kappa} (\Phi_B)^{1-\gamma}, \quad k = 0, 1, \dots, K-1,$$

avec

$$\gamma = 1 - \frac{p+1}{2(p+2)} = \frac{p+3}{2(p+2)}, \quad \bar{\kappa} = \frac{1}{36(p+4)}.$$

Selon (2.5.1), nous obtenons une borne supérieure pour le nombre total d'itérations par l'équation suivante :

$$\frac{72(p+2)(p+4)}{(p+3)} (\Phi_B)_0^{\frac{p+3}{2(p+2)}} \frac{1}{\theta} \log \frac{n}{\varepsilon} \leq 144(p+2) (\Phi_B)_0^{\frac{p+3}{2(p+2)}} \frac{1}{\theta} \log \frac{n}{\varepsilon}. \quad (3.2.20)$$

◆ En utilisant (3.2.15) pour les méthodes à long-pas avec $\tau = O(n)$ et $\theta = \Theta(1)$, on a $(\Phi_B)_0 = O(n)$. Nous distinguons deux cas :

- Si $p \in [1, +\infty[$, on a $O\left(pn^{\frac{p+3}{2(p+2)}} \log \frac{n}{\varepsilon}\right)$ itérations.
- Si $p \in]0, 1]$, on a $O\left(n^{\frac{3}{4}} \log \frac{n}{\varepsilon}\right)$ itérations.

◆ En utilisant (3.2.16) pour les méthodes à court-pas avec $\tau = O(1)$ et $\theta = \Theta(\frac{1}{\sqrt{n}})$.

Nous distinguons deux cas :

- Si $p \in [1, +\infty[$, on a $O(p^2 \sqrt{n} \log \frac{n}{\varepsilon})$ itérations où $(\Phi_B)_0 = O(p)$.
- Si $p \in]0, 1]$, on a $O(\sqrt{n} \log \frac{n}{\varepsilon})$ itérations où $(\Phi_B)_0 = O(1)$.

◆ Si on prend $p = O(\frac{\log n}{2} - 2)$, nous obtenons la meilleure complexité connue pour les méthodes à long-pas, à savoir

$$O\left(\sqrt{n} \log n \log \frac{n}{\varepsilon}\right),$$

qui est le minimum de $O\left((p+2)n^{\frac{p+3}{2(p+2)}} \log \frac{n}{\varepsilon}\right)$.

3.2.3 Tests numériques

Pour prouver l'efficacité de notre nouvelle fonction noyau ψ_B définie dans (3.2.1) et évaluer son effet sur le comportement de l'algorithme, nous allons effectuer des tests numériques comparatifs avec les fonctions noyaux ψ_2 et ψ_4 citées dans le tableau 2.

L'algorithme générique primal-dual décrit dans la section 2.3 a été implémenté en MATLAB 2009, voici quelques clarifications nécessaires dans l'implémentation :

— Nous prenons

$$\varepsilon = 10^{-4}, \mu^0 = 1, \theta \in \{0.01, 0.9\}, \tau = \{n, 10n\}.$$

— Nous utilisons à chaque itération interne :

1. Un pas de déplacement théorique $\alpha_{théor}$, nous prenons pour les trois fonctions noyaux :

$$\alpha_{\psi_2} = \frac{1}{q(4\delta + 1)^{\frac{q+1}{q}}}, \alpha_{\psi_4} = \frac{1}{1 + q(4\delta + 1)^{\frac{q+1}{q}}}, \alpha_{\psi_B} = \frac{1}{2 + (p+2)(4\delta + 2)^{\frac{p+3}{p+2}}},$$

qui correspondent aux fonctions noyaux ψ_2, ψ_4 et ψ_B , respectivement.

2. Un pas de déplacement pratique α_{prat} telle que

$$x^+ = x + \alpha_{prat}\Delta x > 0 \text{ et } z^+ = z + \alpha_{prat}\Delta z > 0.$$

où $\alpha_{prat} = \rho \min(\alpha_x, \alpha_z), \rho \in]0, 1[$, avec

$$\alpha_x = \min \left\{ 1, \min_i \left\{ -\frac{x_i}{\Delta x_i}, \Delta x_i < 0 \right\} \right\}, \alpha_z = \min \left\{ 1, \min_i \left\{ -\frac{z_i}{\Delta z_i}, \Delta z_i < 0 \right\} \right\}.$$

— Nous notons par :

$$\psi_B(t) = \psi_{B,p}$$

$$\psi_2(t) = \psi_{2,q}$$

$$\psi_4(t) = \psi_{4,q}$$

dans les tests, on choisit

$$p = \{\log n\}, q = \left\{ 1.1, 4, 5, \log n, \frac{\log n}{2} \right\}.$$

— Dans les tableaux de résultats : m désigne le nombre de contraintes, n désigne le nombre de variables. Nous indiquons $I_{t\text{ext}}$, $I_{t\text{rint}}$ et $T(s)$ le nombre d'itérations externes, internes et le temps de calculs par seconde respectivement, pour obtenir la solution optimale.

Les exemples sont donnés sous la forme suivante :

$$\begin{cases} \min c^t x + \frac{1}{2} x^t Q x, \\ Ax = b, \\ x \geq 0, \end{cases}$$

On prend les mêmes exemples 1, 2, 3, 4 et 5 dans la section 3.1.3.

Example 6

$$n = 2m, A(i, j) = \begin{cases} 0 & i \neq j \text{ et } j \neq i + m, \\ 1 & i = j \text{ et } j = i + m, \end{cases}$$

$c(j) = -1, c(j + m) = 0, b(j) = 2$ pour $j = 1, \dots, m$. $Q(i, j) = 0$ pour $i, j = 1, \dots, n$.

Le point initial strictement réalisable est :

$$x^0(i) = x^0(i + m) = 1, y^0(i) = -2, z^0(i) = 1, z^0(i + m) = 2 \text{ pour } i = 1, \dots, m.$$

La solution optimale primale-duale obtenue est :

$$x^*(i) = 2, x^*(i + m) = 0, y^*(i) = -1, z^*(i) = 0, z^*(i + m) = 1 \text{ pour } i = 1, \dots, m.$$

Nous résumons les résultats numériques des exemples de 1 à 4 dans les tableaux de 11 à 14 avec $\tau = 10n$. Les résultats numériques de l'exemple 5 et 6 sont présentées dans les tableaux 15, 16 et les tableaux 17, 18 respectivement avec $\tau = n$.

		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$\psi_{2,q=1.1}$	<i>Itrint</i>	2621	3553	5650	10534
	<i>Itrext</i>	1342	1371	1393	1462
	<i>T(s)</i>	4.98	8.17	14.73	50.95
$\psi_{2,q=5}$	<i>Itrint</i>	1493	1836	2704	3979
	<i>Itrext</i>	1331	1360	1382	1451
	<i>T(s)</i>	3.33	5.01	8.17	21.87
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	17972	11226	11759	9571
	<i>Itrext</i>	1349	1376	1396	1461
	<i>T(s)</i>	26.10	21.87	28.25	45.80
$\psi_{4,q=1.1}$	<i>Itrint</i>	2947	4077	6095	11818
	<i>Itrext</i>	1618	1671	1712	1843
	<i>T(s)</i>	5.15	9.02	15.36	55.74
$\psi_{4,q=4}$	<i>Itrint</i>	4430	6196	8754	17010
	<i>Itrext</i>	1603	1658	1701	1836
	<i>T(s)</i>	7.34	12.73	21.20	81.03
$\psi_{4,q=\log n}$	<i>Itrint</i>	2950	3983	5900	12585
	<i>Itrext</i>	1619	1665	1705	1837
	<i>T(s)</i>	5.09	8.58	14.81	58.45
$\psi_{B,p=\log n}$	<i>Itrint</i>	1230	1523	2261	3403
	<i>Itrext</i>	1290	1318	1340	1409
	<i>T(s)</i>	3.03	4.50	7.27	19.63

Tableau 11 : Les résultats numériques des exemples de 1 à 4 avec $\theta=0.01$ et $\alpha=\alpha_{théor}$.

		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$\psi_{2,q=1.1}$	<i>Itrint</i>	8969	11871	22373	40559
	<i>Itrext</i>	6	6	7	7
	<i>T(s)</i>	12.83	22.03	51.09	192.90
$\psi_{2,q=5}$	<i>Itrint</i>	3118	3739	6469	9144
	<i>Itrext</i>	6	6	7	7
	<i>T(s)</i>	4.62	7.05	14.95	42.78
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	131076	71605	60942	35688
	<i>Itrext</i>	6	7	7	7
	<i>T(s)</i>	194.08	137.58	144.86	167.61
$\psi_{4,q=1.1}$	<i>Itrint</i>	6735	8837	12684	26314
	<i>Itrext</i>	8	8	8	9
	<i>T(s)</i>	9.19	15.82	27.88	120.98
$\psi_{4,q=4}$	<i>Itrint</i>	8177	13638	18490	38663
	<i>Itrext</i>	7	8	8	9
	<i>T(s)</i>	11.36	24.47	41.60	178.61
$\psi_{4,q=\log n}$	<i>Itrint</i>	6739	8677	12363	28444
	<i>Itrext</i>	8	8	8	9
	<i>T(s)</i>	9.08	15.37	26.93	127.67
$\psi_{B,p=\log n}$	<i>Itrint</i>	2900	3465	5101	8253
	<i>Itrext</i>	6	6	6	7
	<i>T(s)</i>	4.57	6.99	12.40	38.95

Tableau 12 : Les résultats numériques des exemples de 1 à 4 avec $\theta = 0.9$ et

$\alpha = \alpha_{théor}$.

		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$\psi_{2,q=1.1}$	<i>Itrint</i>	5	5	5	5
	<i>Itrext</i>	1281	1115	1169	1282
	<i>T(s)</i>	1.22	1.37	1.70	3.43
$\psi_{2,q=5}$	<i>Itrint</i>	25	7	7	9
	<i>Itrext</i>	972	1073	1295	1256
	<i>T(s)</i>	1.02	1.38	1.91	3.41
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	5	6	6	5
	<i>Itrext</i>	1190	1342	1354	1297
	<i>T(s)</i>	0.96	1.40	1.75	3.30
$\psi_{4,q=1.1}$	<i>Itrint</i>	271	379	480	923
	<i>Itrext</i>	1617	1669	1711	1842
	<i>T(s)</i>	1.54	2.23	3.09	8.18
$\psi_{4,q=4}$	<i>Itrint</i>	242	348	444	888
	<i>Itrext</i>	1600	1656	1699	1836
	<i>T(s)</i>	1.46	2.23	2.99	7.99
$\psi_{4,q=\log n}$	<i>Itrint</i>	271	365	458	893
	<i>Itrext</i>	1617	1664	1704	1836
	<i>T(s)</i>	1.52	2.23	2.97	7.92
$\psi_{B,p=\log n}$	<i>Itrint</i>	4	5	5	5
	<i>Itrext</i>	969	1179	1237	1343
	<i>T(s)</i>	0.73	1.18	1.53	3.34

Tableau 13 : Les résultats numériques des exemples de 1 à 4 avec $\theta = 0.01$ et $\alpha = \alpha_{prat}$.

		Exemple 1	Exemple 2	Exemple 3	Exemple 4
$\psi_{2,q=1.1}$	<i>Itrint</i>	4	5	5	5
	<i>Itrext</i>	5	5	7	7
	$T(s)$	0.23	0.24	0.25	0.25
$\psi_{2,q=5}$	<i>Itrint</i>	5	6	4	4
	<i>Itrext</i>	5	6	5	6
	$T(s)$	0.23	0.23	0.23	0.25
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	5	5	5	5
	<i>Itrext</i>	6	5	6	7
	$T(s)$	0.04	0.04	0.04	0.07
$\psi_{4,q=1.1}$	<i>Itrint</i>	633	822	1001	2125
	<i>Itrext</i>	8	8	8	9
	$T(s)$	0.90	1.55	2.24	9.30
$\psi_{4,q=4}$	<i>Itrint</i>	445	770	947	2062
	<i>Itrext</i>	7	8	8	9
	$T(s)$	0.64	1.46	2.18	8.90
$\psi_{4,q=\log n}$	<i>Itrint</i>	633	799	969	2075
	<i>Itrext</i>	8	8	8	9
	$T(s)$	0.88	1.44	2.14	9.05
$\psi_{B,p=\log n}$	<i>Itrint</i>	5	5	4	4
	<i>Itrext</i>	6	6	5	5
	$T(s)$	0.02	0.02	0.02	0.04

Tableau 14 : Les résultats numériques des exemples de 1 à 4 avec $\theta = 0.9$ et

$\alpha = \alpha_{prat}$.

	(m, n)	(10, 20)	(30, 60)	(50, 100)	(75, 150)	(100, 200)
$\psi_{2,q=1.1}$	<i>Itrint</i>	3473	10215	16971	25453	33930
	<i>Itrext</i>	1363	1472	1523	1564	1592
	$T(s)$	37.44	445.25	2238.76	6514.19	16205.70
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	2398	4618	6359	8222	9877
	<i>Itrext</i>	1358	1462	1510	1549	1577
	$T(s)$	28.26	185.03	610.53	1459.72	3910.11
$\psi_{4,q=1.1}$	<i>Itrint</i>	2572	7015	11364	16748	22113
	<i>Itrext</i>	1446	1555	1606	1646	1675
	$T(s)$	28.67	294.95	1385.47	4034.62	11654.86
$\psi_{4,q=\log n}$	<i>Itrint</i>	1965	4328	6225	8280	10129
	<i>Itrext</i>	1402	1503	1552	1590	1618
	$T(s)$	25.79	176.74	469.55	1568.95	4021.40
$\psi_{B,p=\log n}$	<i>Itrint</i>	1999	4241	6003	7887	9572
	<i>Itrext</i>	1322	1429	1480	1519	1548
	$T(s)$	25.49	168.58	555.35	1444.03	3814.67

Tableau 15 : Les résultats numériques de l'exemple 5 avec $\theta = 0.01$ et $\alpha = \alpha_{théor.}$

	(m, n)	(5, 10)	(10, 20)	(30, 60)	(50, 100)	(75, 150)
$\psi_{2,q=1.1}$	<i>Itrint</i>	6941	13182	43803	70833	103860
	<i>Itrext</i>	6	6	7	7	7
	$T(s)$	31.74	119.44	2676.98	8912.35	13906.18
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	6320	7256	13311	16979	20797
	<i>Itrext</i>	6	6	7	7	7
	$T(s)$	27.95	66.03	444.16	1248.50	3460.50
$\psi_{4,q=1.1}$	<i>Itrint</i>	2884	6000	15307	27855	40084
	<i>Itrext</i>	6	7	7	8	8
	$T(s)$	12.47	52.42	529.80	2140.06	8281.39
$\psi_{4,q=\log n}$	<i>Itrint</i>	2374	4437	9070	12613	16347
	<i>Itrext</i>	6	7	7	7	7
	$T(s)$	11.55	41.25	393.39	1502.30	3366.94
$\psi_{B,p=\log n}$	<i>Itrint</i>	2794	4254	9766	13285	16941
	<i>Itrext</i>	6	6	7	7	7
	$T(s)$	13.16	40.71	362.69	1170.33	3032.32

Tableau 16 : Les résultats numériques de l'exemple 5 avec $\theta = 0.9$ et $\alpha = \alpha_{théor}$.

	(m, n)	(5, 10)	(10, 20)	(30, 60)	(50, 100)	(75, 150)	(100, 200)
$\psi_{2,q=1.1}$	<i>Itrint</i>	1782	3472	10225	16971	25452	33930
	<i>Itrext</i>	1294	1363	1472	1523	1564	1592
	$T(s)$	11.32	38.30	405.48	1832.75	4760.53	14712.98
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	1688	2398	4618	6359	8222	9788
	<i>Itrext</i>	1293	1358	1462	1510	1549	1577
	$T(s)$	10.88	27.92	184.18	516.98	1588.73	3610.71
$\psi_{4,q=1.1}$	<i>Itrint</i>	1408	2570	7015	11364	16747	22112
	<i>Itrext</i>	1376	1445	1555	1606	1646	1675
	$T(s)$	10.25	32.45	271.61	802.33	2962.39	9202.87
$\psi_{4,q=\log n}$	<i>Itrint</i>	1193	1963	4328	6225	8279	10129
	<i>Itrext</i>	1341	1401	1503	1552	1590	1618
	$T(s)$	8.54	24.79	244.40	570.81	1480.18	3922.64
$\psi_{B,p=\log n}$	<i>Itrint</i>	1240	1999	4241	6002	7887	9571
	<i>Itrext</i>	1254	1322	1429	1480	1519	1548
	$T(s)$	9.01	26.45	182.90	484.91	1454.79	3401.88

Tableau 17 : Les résultats numériques de l'exemple 6 avec $\theta = 0.01$ et $\alpha = \alpha_{théor.}$

	(m, n)	(5, 10)	(10, 20)	(30, 60)	(50, 100)	(75, 150)
$\psi_{2,q=1.1}$	<i>Itrint</i>	6944	13164	43800	70849	103872
	<i>Itrext</i>	6	6	7	8	7
	$T(s)$	31.20	126.49	2452.91	8441.02	21012.14
$\psi_{2,q=\frac{\log n}{2}}$	<i>Itrint</i>	6332	7247	13306	16978	20802
	<i>Itrext</i>	6	6	7	7	7
	$T(s)$	28.03	64.16	481.82	1782.17	4393.51
$\psi_{4,q=1.1}$	<i>Itrint</i>	2882	5997	15307	27854	40083
	<i>Itrext</i>	6	7	7	8	8
	$T(s)$	13.13	55.64	586.75	2068.99	6175.48
$\psi_{4,q=\log n}$	<i>Itrint</i>	2374	4437	9071	12613	16347
	<i>Itrext</i>	6	7	7	7	7
	$T(s)$	10.65	42.06	493.95	1043.59	3120.93
$\psi_{B,p=\log n}$	<i>Itrint</i>	2791	4252	9761	13283	16940
	<i>Itrext</i>	6	6	7	7	7
	$T(s)$	12.89	41.50	368.59	1022.30	2948.33

Tableau 18 : Les résultats numériques de l'exemple 6 avec $\theta = 0.9$ et $\alpha = \alpha_{théor}$.

Commentaires

A travers les tests numériques qu'on a effectués, nous déduisons ce qui suit :

- Pour les exemples de 1 à 4, le pas de déplacement pratique α_{prat} du tableaux 13 et 14 offre de meilleurs résultats que la pas théorique du tableaux 11 et 12 en nombre d'itérations et en temps de calculs.
- Les expérimentations numériques réalisées montrent l'efficacité de notre nouvelle fonction noyau ψ_B . Nous notons que lorsque la dimension du problème devient grande, la différence entre notre nouvelle fonction noyau ψ_B , celle de Peng et al. [60] ψ_2 et celle de Bai et al. [7, 9] ψ_4 lorsque $\theta = 0.01$ devient grande en termes de nombre total d'itérations et de temps de calculs mettant en faveur notre nouvelle fonction ψ_B et en temps de calculs lorsque $\theta = 0.9$.
- Le nombre d'itérations externes est considérablement réduit lorsque $\theta = 0.9$ pour tous les exemples testés.

Ces résultats numériques consolident et confirment nos résultats théoriques.

Conclusion 3.2.9. Dans ce chapitre, nous avons proposé deux nouvelles fonctions noyaux paramétrées qui ne sont ni logarithmiques ni auto-régulières :

- La première fonction noyau proposée a un terme barrière exponentiel. Nous avons prouvé que la borne d'itération de la méthode de points intérieurs basée sur cette fonction noyau est $O\left(\sqrt{n}(\log n)^2 \log \frac{n}{\varepsilon}\right)$ pour la méthode à long-pas et $O\left(\sqrt{n} \log \frac{n}{\varepsilon}\right)$ pour la méthode à court-pas qui est la meilleure borne de complexité connue j'ausqu'à présent. Notre approche a permis de généraliser le résultat obtenu par Bai et al. [7] et d'améliorer le comportement pratique de notre algorithme (**Algorithme 5**) qui est présenté dans la section 2.3.

- La deuxième nouvelle fonction noyau consiste en un polynôme dans son terme barrière. Une analyse simple pour les MPIs primales-duales basées sur la fonction barrière induite par la nouvelle fonction noyau est fournie, nous avons prouvé que la borne d'itérations de la méthode de points intérieurs basée sur cette fonction noyau est $O\left(\sqrt{n} \log n \log \frac{n}{\varepsilon}\right)$ pour la méthode à long-pas et $O\left(\sqrt{n} \log \frac{n}{\varepsilon}\right)$ pour la méthode à court-pas. Cette borne améliore de manière significative les résultats obtenus par Peng et al. dans [60] et Bai et al. dans [9].

CONCLUSION GÉNÉRALE

Dans cette thèse, nous nous sommes intéressés à la résolution d'un programme quadratique convexe avec contraintes linéaires (PQC). Plusieurs problèmes concrets dans plusieurs domaines peuvent se formuler sous forme de PQC , c'est l'intérêt de résoudre ce problème.

Les méthodes de points intérieurs et plus précisément les méthodes de trajectoire centrale sont les plus efficaces pour résoudre ce type des problèmes. L'inconvénient de ces méthodes est le problème d'initialisation pour démarrer leur algorithme en pratique. Pour surmonter cette difficulté, nous avons présenté dans cette thèse les bases théoriques et algorithmiques d'une alternative particulièrement intéressante : il s'agit des $MPIs$ basées sur les fonctions noyaux qui ont l'avantage de démarrer avec un point de départ strictement réalisable non nécessairement proche de la trajectoire centrale.

D'autre part, nous avons proposé deux nouvelles fonctions noyaux paramétrées. La première fonction noyau avec terme barrière exponentiel généralise celle donnée par Bai et al. [7]. La deuxième fonction noyau paramétrée a un terme barrière polynomial (de Laurent) qui donne la meilleure complexité obtenue jusqu'à présent pour certaine valeur du paramètre p et elle améliore de manière significative les résultats obtenus par Peng et al. dans [60, 61] et Bai et al. dans [9].

Ces travaux sont des contributions importantes permettant d'améliorer la complexité des algorithmes et le comportement numérique des méthodes de points intérieurs primales-duales pour PQC . Les résultats obtenus sont publiés dans deux revues internationales [18, 19].

CONCLUSION GÉNÉRALE

Nos perspectives est de généraliser nos travaux pour résoudre d'autres problèmes d'optimisation tel que : les problèmes convexes non nécessairement quadratiques avec contraintes linéaires.

BIBLIOGRAPHIE

- [1] M. Achache, A new primal-dual path-following method for convex quadratic programming. *Comput. Appl. Math.* 25(1), 97–110, (2006).
- [2] M. Achache, A new parameterized kernel function for LO yielding the best known iteration bound for a large-update interior point algorithm. *Afrika Matematika*, 27 (3), 591-601, (2016).
- [3] M. Achache and M. Goutali, A primal-dual interior point algorithm for Convex Quadratic Programs. *Studia Univ. Babes-Bolyai. Ser. Inform.* LVII (1), 48–58, (2012).
- [4] M. Achache , H. Roumili, A. Keraghel, A numerical study of an infeasible primal-dual path-following algorithm for linear programming. *Applied Mathematics and Computation* 186, 1472–1479, (2007).
- [5] M. Achache, N. Tabchouche, Complexity analysis and numerical implementation of large-update interior point methods for SDLCP based on a new parametric kernel function. *Optimization.* 67 (8), 1211-1230, (2018).
- [6] E.D. Andersen, J. Gondzio, Cs. Mészáros, X. Xu, Implementation of interior point methods for large scale linear programming, in *Interior Point Methods of Mathematical Programming*, Terlaky T., ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 189–252, (1996).
- [7] Y.Q. Bai, M. El Ghami, C. Roos, A comparative study of kernel functions for primal-dual interior point algorithms in linear optimization. *SIAM. J. Optim.* 15(1), 101–128, (2004).

- [8] Y.Q. Bai, G. Lesaja, C. Roos, G.Q. Wang, and M. ElGhami, A class of large- and small-update primal-dual interior-point algorithm for linear optimization. *Journal of optimization theory and application*, 138(3), 341-359, (2008).
- [9] Y.Q. Bai, C. Roos, A primal-dual interior point method based on a new kernel function with linear growth rate, In : *Proceedings of the 9th Australian Optimization Day, Perth, Australia*. 1 (2002).
- [10] E.R. Barnes, Variation of Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, vol 36, 174-182, (1986).
- [11] D. Bayer and J-C. Lagarias, The nonlinear geometry of linear programming. II legendre transform coordinates and central trajectories *Transactions, the American Mathematical Society*, 314(2), 27-581, (1989).
- [12] S. Bazarra, H.D. Sherali and C.M. Shetty, *Nonlinear programming, theory and algorithms*, Second edition, (1993).
- [13] M. Bouafia, *Etude asymptotique des méthodes de points intérieurs pour la programmation linéaire*, Thèse de doctorat, Université du Havre (2016).
- [14] M. Bouafia, D. Benterki, A. Yassine, Complexity analysis of interior point methods for linear programming based on a parameterized kernel function, *RAIRO-Oper. Res.* 50, 935–949, (2016).
- [15] M. Bouafia, D. Benterki, A. Yassine, An efficient primal-dual interior point method for linear optimization based on a new kernel function with a trigonometric barrier term, *J. Optim Theory App* 170(5), 528–545, (2016).
- [16] M. Bouafia, D. Benterki, A. Yassine, An efficient parameterized logarithmic kernel function for linear optimization, *Optim Lett* 12(5), 1079–1097, (2018).
- [17] M. Bouafia, A. Yassine, An efficient twice parameterized trigonometric kernel function for linear optimization, *Optimization and Engineering. Optim Eng* 21, 651–672, (2020).
- [18] N. Boudjellal, H. Roumili, Dj. Benterki, A primal-dual interior point algorithm for convex quadratic programming based on a new parametric kernel function. *Optimization*. DOI : 10.1080/02331934.2020.1751156, (2020).

- [19] N. Boudjellal, H. Roumili, Dj. Benterki, Complexity Analysis of Interior Point Methods for Convex Quadratic Programming Based on a Parameterized Kernel Function. *Bol. Soc. Paran. Mat.* (Accepted).
- [20] M.A. Boughazi, Contribution à l'étude des algorithmes d'optimisation en analyse des données, Thèse de doctorat, Université de J. Fourier Grenoble 1, (1987).
- [21] T.M. Cavalier and AL. Soyster, Some computational experience and a modification of Karmarkar algorithm. Presented ut the 12th International Symposium on Mathematical Programming, Massachusetts Institute of Technology, Cambridge, Massachsets, August, (1985).
- [22] X.Z. Cia, G.Q. Wang, M. El Ghami, Y.J.Yue, Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term. *Abstr. Appl. Anal.*, Art. ID 710158, 11, (2014).
- [23] I. S. Ciupercai, Cours optimisation en Master M1 SITN.
- [24] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, (1963).
- [25] L. Derbal, Z. Kebbiche, Theoretical and Numerical Result for Linear Optimization Problem Based on a New Kernel Function, *Journal of Siberian Federal University. Mathematics & Physics*, 12(2), 160–172, (2019).
- [26] E. A. Djeflal, Etude de quelques algorithmes de points intérieurs pour la programmation convexe, Thèse de doctorat d'état, Université Hadj Lakhdar. Batna, (2013).
- [27] I.I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics Dokhdy*, vol 8, 674-675, (1967).
- [28] M. EL GHAMI, New Primal-dual Interior-point Methods Based on Kernel Functions, Thèse de doctorat, Universite Mohammed V, Rabat, Marokko, (2005).
- [29] M. El Ghami, Z.A. Guennoun, S. Bouali, T. Steihaug, Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term. *J. Comput. Appl. Math.* 236, 3613–3623, (2012).

- [30] M. EL Kharboutly, Résolution d'un problème quadratique non convexe avec contraintes mixtes par les techniques de l'optimisation D.C., Thèse de doctorat, Université Le Havre Normandie, (2018).
- [31] A. V. Fiacco, G. P. McCoRMicK, Nonlinear programming : Sequential unconstrained minimization techniques, John Wiley, a New York, (1968).
- [32] R. Fletcher, Practicam methods of optimization, NewYork, john Wiley & sons edition, (1991).
- [33] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly, vol 8, 95-110, (1956).
- [34] K. R. Frisch, The logarithmic potential method of convex programming. Memorandum of May 13 (1955), University Institute of Economics, Oslo, Norway.
- [35] P. Hammer, P. Hansen, B. Simeone, Roof duality complementation and persistency in quadratic 0-1 optimization, Mathematical programming, vol 28, 121-155, (1984).
- [36] M.R. Hestenes, Multiplier and gradient methods. Journal of optimization theory and application, 303-320, (1969).
- [37] R. Herbin, Cours d'Analyse numérique Licence de mathématiques, Université Aix Marseille, (2014).
- [38] C.C. Gonzaga, Polynomial affine algorithm for linear programming. Mathematical Programming, 49, 7-21, (1990).
- [39] L. Guerra, Méthodes de points intérieurs et fonctions noyaux pour l'optimisation quadratique semi-définie convexe, Thèse de doctorat, Université Ferhat Abbas, Sétif-1, (2018).
- [40] E. KALLEL, Une synthèse sur les méthodes du point intérieur, Faculté des sciences université de SHERBROOKE, Canada, (1998).
- [41] S. Karlin, Mathematical methods and theory in games, programming and economies, vol 1, Addison Wesley, (1959).
- [42] N. Karmarkar, A new polynomial-time algorithm for linear programming. In Proceeding of the 16th Annual ACM Symposium on Weory of computing, 303-311, (1984).

- [43] Z. Kebbiche, Etude et extensions d'algorithmes de points intérieurs pour la programmation non linéaire. Thèse de doctorat d'état, Université Ferhat Abbas - Sétif 1, (2007).
- [44] A. Keraghel, Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar, Thèse de doctorat, Université Joseph Fourier-Grenoble I, (1989).
- [45] L. G. Khachian, A polynomial algorithm in linear programming. Soviet Mathematics Doklady, 20, 191-194, (1979).
- [46] B. Kheirfam, M. Moslem, A polynomial-time algorithm for linear optimization based on a new kernel function with trigonometric barrier term. YUJOR 25(2), 233-250, (2015).
- [47] Klee, Minty, How good is the simplex algorithm?, Inequalities. O. Shisha, editor, Academic Press, New York, 159-175, (1972).
- [48] M. Kojima, S. Mizuno, and A. Yoshise, A polynomial-time algorithm for a class of linear complementarity problems. Mathematical Programming, 44 : 1-26, (1989).
- [49] M. Kojima, S. Mizuno & A. Yoshise, A primal-dual interior point algorithm for linear programming. Progress in Mathematical Programming : Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 29-47, (1989).
- [50] H.A. Le Thi, An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints, Mathematical programming, vol 87(3), 401-426, (2000).
- [51] X. Li, M. Zhang, Interior-point algorithm for linear optimization based on a new trigonometric kernel function. Oper. Res. Lett 43(5), 471-475, (2015).
- [52] Z.S. Lu, R.D.C. Monteiro, J.W. O'Neal, An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming. SIAM J. Optim. 17(1), 287-310, (2006).
- [53] S. Mehrotra, On an implementation of primal-dual predictor-corrector algorithms, presented at the Second Asilomar Workshop on Progress in Mathematical Programming, Asilomar, CA. (1990).

- [54] N. Meggido, Introduction : New approaches to linear programming. *Algorithmica*, 1 (4) : 387-394, (1986).
- [55] B. Merikhi, Etude comparative de l'extension de l'algorithme de Karmarkar et des méthodes simpliciales pour la programmation quadratique convexe, Thèse de magister, Université de Ferhat Abbas Sétif, (1994).
- [56] M. Minoux, Programmation mathématique, théorie et algorithmes, tome 1, 112-123, Dunod Paris, (1983).
- [57] R.D.C. Monteiro and I. Adler, Interior path following primal-dual algorithms. part 1 : Linear programming. *Mathematical Programming*, 44 : 27-41 (1989).
- [58] R.D.C. Monteiro and I. Adler, Interior path-following primal-dual algorithms : Part II : Convex programming, *Mathematical Programming*, 44 : 43-66, (1989).
- [59] J. Peng, C. Roos, T. Terlaky, New complexity analysis of the primal-dual Newton method for linear optimization, *Anna Oper. Res.* 49, 23-99, (2000).
- [60] J. Peng, C. Roos, T. Terlaky, A new and efficient large-update interior point method for linear optimization. *J. Comput. Technol.* 6, 61-80, (2001).
- [61] J. Peng, C. Roos, T. Terlaky, Self-Regularity, A New Paradigm for Primal-Dual Interior Point Algorithms. Princeton University Press, Princeton, (2002).
- [62] J. Peng, C. Roos, T. Terlaky & A. Yoshise, Self-Regular proximities and new directions for nonlinear $P_*(k)$ complementarity problems, *Math. Oper. Res.*, (2002).
- [63] M.R. Peyghami, S.F. Hafshejani, L., Shirvani, Complexity of interior point methods for linear optimization based on a new trigonometric kernel function. *J. Comput. Appl. Math.* 255, 74-85, (2014).
- [64] R.T. Rockfellar, *Convex Analysis*. Princeton University Press, N.J., (1970).
- [65] H. Roumili, Méthodes de points intérieurs non réalisables en optimisation. Théorie, Algorithme et Applications., Thèse de doctorat sciences, Université Ferhat Abbas- Sétif 1, (2007).
- [66] H. Roumili, Short-step primal-dual target-following algorithms for the convex quadratic problems. *International Journal of Pure and Applied Mathematics*, Volume 77 No. 3, 337-348, (2012).

- [67] H. Roumili, Weighted-path-following algorithm for linear programming. *Math. Reports* 15(65), 145-152 (2013).
- [68] H. Roumili, N. Boudjellal, An infeasible interior point methods for convex quadratic problems. *J. Numer. Anal. Approx. Theory*, 47(2), 177-186, (2019).
- [69] H. Roumili, A. Keraghel, and A. Yassine, Infeasible primal-dual algorithm for minimizing convex quadratic problems, *STUDIA UNIV. BABE BOLYAI, INFORMATICA*, Volume XLIX, Number 2, (2004).
- [70] C. Roos, T. Terlaky, and J.-P. Vial, *Theory and Algorithms for Linear Optimization. An Interior-Point Approach*. Springer Science, (2005).
- [71] C. Roos and J.-Ph. Vial, A polynomial method of approximate centers for linear programming. *Mathematical Programming*, 54 : 295-305, (1992).
- [72] M. SLATER, Lagrange Multipliers Revisited : A Contribution to Nonlinear a Programming, Cowles Commission Discussion Paper, Mathematics 403, (1950).
- [73] M.J. Todd and B.P. Burrell, An extension of Karmarkar's algorithm for linear programming using dual variables. *Algorithmica*, 1, 409-424, (1986).
- [74] M.J. Todd and Y. Ye, A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 2 : 98-209, (1990).
- [75] P. Tseng and Z.Q. Luo, On the convergence of the affine-scaling algorithm. *Mathematical Programming*, 56, 01-319, (1992).
- [76] R.J. Vanderbei, M.S. Meketon. and B.A. Freedman, Modification of Karmarkar's linear programming algorithm. *Algorithmica*, 4(1) : 395-408, (1986).
- [77] G.Q. Wang, Y.Q. Bai, Y. Liu, M. Zhang, A new primal-dual interior-point algorithm for convex quadratic optimization. *J. Shanghai Univ.* 12(3), 189-196, (2008).
- [78] S.J. Wright, *Primal-dual interior point methods*, Copyright by SIAM, (1997).
- [79] Y. Ye, An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50, 239-258, (1991).
- [80] Y. Ye, *Interior-Point Methods for Linear Optimization*. Springer, New York, USA (2006).

- [81] Y. Ye and M. Kojima, Recovering optimal basis is Karmarkar's polynomial algorithm for linear programming. *Mathematical Programming*, 39, 305-317, (1987).
- [82] Y. Yinu and T. Edison, An extension of Karmarkar projective algorithm for convex quadratic programming. *Mathematical Programming*, 40, (1987).
- [83] Q. Yu, C.C. Huang, Y. Jiang, A polynomial predictor-corrector interior-point algorithm for convex quadratic programming. *Acta Math. Sci. Ser. B Engl. Ed.* 26B(2), 263-270, (2006).

ملخص:

في هذه الأطروحة، يتم تقديم فئة من طرق النقاط الداخلية الأولية-الثنوية (IPMs) لحل مسائل البرمجة التربيعية المحدبة. إنها طريقة مسار مركزي المعتمدة على دوال النواة المقترحة بهدف معالجة مسألة النقاط الابتدائية (النقطة الابتدائية تكون في جوار المسار المركزي) من خلال إنشاء مرحلة المركزية التي تقاس بدالة الحاجز. نقترح دالتين جديدتين للنواة ذات معلمات الأولى لها مصطلح حاجز أسي والثانية لها مصطلح حاجز كثير الحدود. نقوم بتحليل طرق الخطوة الكبيرة والصغيرة التي تعتمد على دوال النواة الجديدة. نحصل على أفضل حدود التكرار المعروفة فيما يتعلق بطرق الخطوة الصغيرة للدالتين نواة و فيما يتعلق بطرق الخطوة الكبيرة للدالة النواة الثانية. وأخيراً يتم تقديم بعض النتائج العددية لإظهار كفاءة وظائف دوال النواة المقترحة.

كلمات مفتاحية: طرق النقاط الداخلية الأولية-الثنوية، البرمجة التربيعية المحدبة، طرق الخطوة الكبيرة والصغيرة، دوال النواة.

Abstract:

In this thesis, a class of primal-dual interior point methods (IPMs) for solving convex quadratic programming problems is presented. It is a central path method based on a kernel function which is proposed in order to remedy the initialization problem (the initial point is in the neighborhood of central path) by creating the phase of centrality which is measured by barrier function. We propose two new parameterized kernel functions. The first one has an exponential barrier term and the second has a polynomial barrier term. We analyze large and small-update versions which are based on these new kernel functions. We obtain the best-known iteration bounds concerning the small-update version for the two kernel functions and the large-update version for the second kernel function. Finally, some numerical results are presented to show the efficiency of the proposed kernel functions.

Keywords: Primal-dual interior point methods, Convex quadratic programming, Large and small-update versions, Kernel function.

Résumé :

Dans cette thèse, une classe de méthodes de points intérieurs primales-duales (MPIs) pour résoudre des problèmes de programmation quadratique convexe est présentée. C'est une méthode de trajectoire centrale basée sur une fonction noyau qui est proposée dans le but de remédier au problème d'initialisation (le point initial soit au voisinage de la trajectoire centrale) en créant la phase de centralité qui est mesurée par fonction barrière. Nous proposons deux nouvelles fonctions noyaux paramétrées. La première a un terme barrière exponentiel et la seconde a un terme barrière polynomial. Nous analysons les versions à grand et petit pas qui sont basées sur ces nouvelles fonctions noyaux. Nous obtenons les meilleures bornes d'itérations connues concernant la petite version pour les deux fonctions noyaux et la grande version de mise à jour pour la deuxième fonction noyau. Enfin, quelques résultats numériques sont présentés pour montrer l'efficacité des fonctions noyaux proposées.

Mots clés : Méthodes de points intérieurs primales-duales, Programmation quadratique convexe, Version à grand et petit pas, Fonction noyau.