

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 :

DERBAL LOUIZA
Pour obtenir le titre de Doctorat en Sciences

OPTION
MATHÉMATIQUES APPLIQUÉES

THÈME
Application des méthodes de points intérieurs pour
certains problèmes semi-définis :
Théorie et algorithmes

Soutenue le : 04 / 12 / 2019

Devant le jury composé de :

Président :	Mr. M. ACHACHE	Prof.	U.F.A. Sétif-1
Encadreur :	Mme. Z. KEBBICHE	M.C.A.	U.F.A. Sétif-1
Examineurs :	Mr. M. ZERGUINE	M.C.A.	U.M.B. Batna-2
	Mr. M. BOUAFIA	M.C.A.	U.8 Mai 1945 Guelma

Année universitaire 2018-2019

REMERCIEMENTS

Dieu merci pour cette réussite et ce succès, pour le courage et la patience que vous m'avez accordés le long de mon parcours de formation surtout.

*Un grand remerciement à mon encadreur Mme **Zakia KEBBICHE**, Maître de Conférences classe A à l'université -Ferhat Abbas- Sétif 1, pour ses consignes, ses conseils fructueux et sa disposition. Sans sa patience, ce travail n'aurait pas pu voir le jour.*

*J'exprime mon respect à Monsieur **Mohamed ACHACHE**, Professeur à l'université -Ferhat Abbas- Sétif 1, qui m'a fait l'honneur de présider le jury de cette thèse que j'ai pu améliorer grâce à ses conseils pertinents, ses remarques et ses encouragements.*

Je tiens aussi à exprimer mes vifs remerciements et mes sincères respects à :

*Monsieur **Mohamed ZERGUINE**, Maître de conférences classe A à l'université de Batna 2 -Mostefa Ben Boulaid, pour l'attention toute particulière qu'il a accordée à ce travail et qui a accepté d'en être un examinateur extérieur.*

*Monsieur **Mousâb BOUAFIA**, Maître de Conférences classe A à l'université -8 Mai 1945- Guelma, pour l'intérêt qu'il a porté à ce travail et pour son amabilité d'avoir bien voulu participer au jury.*

Mes sincères remerciements vont aussi à tous les membres de l'équipe d'optimisation de l'université Ferhat Abbas sans oublier de les adresser à l'équipe administrative de l'institut de mathématiques, à tous les membres du conseil scientifique, à tous les enseignants et mes amies.

Je tiens à remercier particulièrement ma famille qui m'a toujours soutenu et encouragé sur ce chemin : le chemin du savoir

Table des matières

0.1	Introduction	4
1	Rappels de base	12
1.1	Analyse convexe	12
1.2	Analyse matricielle	14
1.3	Programmation mathématique	18
1.3.1	Existence et unicité de solution	19
1.3.2	Conditions d'optimalité	19
1.3.3	Dualité d'un programme mathématique	20
1.3.4	Algorithme d'optimisation	21
1.3.5	Programmation linéaire	22
1.3.6	Dualité en programmation linéaire	23
2	Méthodes de points intérieurs	26
2.1	Méthodes de trajectoire centrale (TC) pour la programmation linéaire . .	28
2.1.1	Méthodes de trajectoire centrale classique	28
2.1.2	Méthodes de trajectoire centrale avec poids	31
2.1.3	Méthodes de trajectoire centrale via une fonction noyau	36
2.2	Méthodes de trajectoire centrale pour la programmation semi-définie . .	43
2.2.1	Dualité en programmation semi-définie	46
2.2.2	Complémentarité en <i>SDP</i>	50

2.2.3	Conditions de Karush - Kuhn - Tucker (<i>KKT</i>)	51
2.2.4	Fonction barrière	51
2.2.5	Trajectoire centrale et mesure de dualité	52
2.2.6	Description de la méthode de <i>TC</i>	53
3	Méthodes de (TC) pour (PL) et (SDP) basées sur une nouvelle fonction noyau	59
I	Programmation linéaire	61
3.1	Propriétés de la nouvelle fonction noyau	62
3.2	Analyse de la complexité de l'algorithme	68
3.3	Expérimentations numériques	78
II	Extension au problème semi-défini linéaire	84
3.4	Description de l'algorithme	85
3.4.1	Borne supérieure de $\Psi(V)$ pour chaque itération externe	86
3.4.2	Décroissance de $\Psi(V)$ durant une itération interne	87
3.5	Expérimentations numériques	90
3.6	Conclusion	97

0.1 Introduction

Les méthodes de points intérieurs constituent le sujet de recherche le plus large dans le domaine de l'optimisation avec contraintes. Elles sont nées juste après l'apparition de l'algorithme du simplexe (1947-1951), dans le but de résoudre le problème "théorique" concernant la complexité arithmétique "exponentielle" de cet algorithme, exprimée par le nombre total de sommets, qu'il doit visiter éventuellement. Leurs algorithmes calculent une suite d'itérés appartenant à l'intérieur de l'espace admissible et convergeant vers la solution optimale. Ces méthodes ont été très rapidement populaires car elles offrent la possibilité de résoudre un problème d'optimisation linéaire en un temps polynômial dans le pire des cas. Ainsi, elles offrent la promesse de réduire de manière significative le temps de résolution des problèmes de la programmation linéaire. Un concept majeur sur lequel sont fondées les méthodes de points intérieurs consiste à remplacer la résolution du problème d'optimisation linéaire initial par celle d'un système non linéaire d'équations exprimant les conditions d'optimalité de Karush-Kuhn-Tucker. Ceci implique l'utilisation de la méthode de Newton pour la résolution de ce système d'équations non linéaires. Plusieurs méthodes sont élaborées, parmi lesquelles :

- La méthode de Dikin (1967) restée inconnue à l'époque.
- La méthode de pénalité intérieure de type barrière logarithmique de Fiacco et McCormick (1968) [30].
- La méthode des ellipsoïdes, de Khatchian (1979) [40].

En général, les méthodes de cette époque possèdent de bonnes propriétés théoriques, néanmoins, leurs comportements numériques sont avérés non satisfaisants. L'apparition de l'algorithme de Karmarkar (1984)[36] a donné aux méthodes de points intérieurs une nouvelle tournure caractérisée essentiellement par un comportement numérique rivalisant celui des algorithmes fondamentaux dans ce domaine. Bien que le domaine d'optimisation linéaire soit alors considéré comme plus ou moins mûr, il ressort soudainement du papier de Karmarkar comme l'un des domaines de recherche les plus actifs en optimisation.

Bientôt, il devient évident que cette méthode était liée aux méthodes classiques comme la méthode de Dikin [24, 25, 26], la méthode barrière logarithmique de Frisch [31, 32], et la méthode du centre de Huard [35], et que les deux dernières méthodes, une fois réglées correctement, pourraient également être prouvées polynomiales. De plus, il s'est avéré que l'approche des méthodes de points intérieurs à la programmation linéaire a une généralisation naturelle à l'optimisation non linéaire dans le domaine convexe, qui a abouti à un nouveau flux de recherche et à une excellente monographie de Nesterov et Nemirovski [43]. Cette monographie a ouvert le chemin dans d'autres nouveaux sous-domaines de l'optimisation, comme l'optimisation semi-définie et l'optimisation conique, avec des applications importantes dans la théorie des systèmes, de l'optimisation discrète, et de nombreux autres domaines. Pour un aperçu de ces développements le lecteur peut consulter Vandenberghe et Boyd [21], le livre de Ben-Tal et Nemirovski [14], Pen et al. [46], Roos et al. [51] et Wright [56].

La plupart de ces plus récents travaux se sont concentrés sur les méthodes primales-duales. On distingue trois classes de ces méthodes :

- Méthodes affines.
- Méthodes de réduction de potentiel.
- Méthodes de trajectoire centrale.

Notre travail apporte des contributions théoriques, algorithmiques et numériques à la résolution des problèmes d'optimisation de la programmation linéaire et semi-définie linéaire. On s'intéresse plus particulièrement, aux méthodes de type trajectoire centrale primale-duale, car, d'une part, ces algorithmes admettent un bon comportement théorique telle que la complexité polynômiale et d'autre part, ils sont de type Newton ce qui conduit à une efficacité numérique. La majorité de ces méthodes utilisent la fonction logarithmique classique comme étant une fonction barrière. Mais, actuellement il existe un écart entre le comportement pratique et théorique, surtout pour les algorithmes à grand pas pour lesquels la borne théorique d'itérations est d'ordre $\mathcal{O}(n \log \frac{n}{\epsilon})$. En pratique, les algorithmes à grand pas sont plus efficaces que les algorithmes à petit pas pour lesquels

la borne théorique d'itérations n'est que $\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$. En 2002, Peng et al. [46] ont introduit de nouvelles fonctions barrières "fonctions auto-régulières" pour les méthodes de points intérieurs primales-duales appliquées aux problèmes linéaires et récemment, les chercheurs, dans leurs articles [10, 12, 29, 22, 13, 41, 57] et [27, 48, 49, 18, 28] (ordonnés chronologiquement) ont proposé une autre nouvelle classe de méthodes de points intérieurs toujours pour la programmation linéaire et semi-définie linéaire basée sur des fonctions noyaux qui ne sont pas logarithmiques et pas nécessairement auto-régulières [45]. Ceci a permis d'améliorer tous les résultats théoriques obtenus jusqu'à présent et de fournir la meilleure complexité algorithmique surtout aux méthodes à grand pas.

Dans le tableau suivant, on présente quelques exemples de fonctions noyaux avec les résultats de complexité qui correspondent aux méthodes à grand et à petit pas.

i	Fonction noyau $\psi_i(t)$	Méthode à grand pas	Méthode à petit pas	Références
1	$\frac{t^2-1}{2} - \log t$	$\mathcal{O}(n \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[51] Roos et al.(1997)
2	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q(q-1)} - \frac{q-1}{q}(t-1), q > 1$	$\mathcal{O}(qn^{\frac{q+1}{2q}} \log \frac{n}{\epsilon})$	$\mathcal{O}(q^2 \sqrt{n} \log \frac{n}{\epsilon})$	[45] Peng et al.(2003)
3	$\frac{t^2-1}{2} + \frac{1}{b}(e^{b(1-t)} - 1), b > 0$	$\mathcal{O}(\sqrt{n} \log n \log \frac{n}{\epsilon})$ si $b = \log n$	–	[10] Bai et al.(2003)
4	$\frac{1}{2}(t - \frac{1}{t})^2$	$\mathcal{O}(n^{\frac{2}{3}} \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[12] Bai et al.(2004)
5	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}, q > 1$	$\mathcal{O}(qn^{\frac{q+1}{2q}} \log \frac{n}{\epsilon})$	$\mathcal{O}(q^2 \sqrt{n} \log \frac{n}{\epsilon})$	/
6	$\frac{t^2-1}{2} + \frac{e^{\frac{1}{t}} - e}{e}$	$\mathcal{O}(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	/
7	$(m+1)t^2 - (m+2)t + \frac{1}{m}$ $t > 0, m > 4$	$\mathcal{O}(m^{\frac{3m+1}{2m}} n^{\frac{m+1}{2m}} \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[41] Liu et al.(2011)
8	$\frac{t^2-1}{2} - (t-1)e^{\frac{1}{t}-1}$	$\mathcal{O}(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon})$	–	[57] Zhang (2012)
9	$\frac{t^2-1}{2} + \frac{6}{\pi} \left(\frac{\pi(1-t)}{2+4t} \right)$	$\mathcal{O}(n^{\frac{3}{4}} \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[27] El Ghami et al.(2012)
10	$\frac{t^2-1}{2} - \log t + \frac{1}{8} \tan^2 \left(\frac{\pi(1-t)}{2+4t} \right)$	$\mathcal{O}(n^{\frac{3}{4}} \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[48] Peyghami et al.(2014)
11	$\frac{t^2-1}{2} - \int_1^t e^{3 \left(\tan \left(\frac{\pi}{2+2\xi} \right) \right)^{-1}} d\xi$	$\mathcal{O}(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon})$	$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$	[49] Peyghami et al.(2014)
12	$t^2 - 1 + \frac{t^{1-q_1} - 1}{q_1 - 1} + \frac{t^{1-q_2} - 1}{q_2 - 1},$ $q_i > 1, i=1,2.$	$\mathcal{O}((q_1 + 1)n^{\frac{q_1+1}{2(q_1-q_2)}} \log \frac{n}{\epsilon}),$ $q_1 > q_2 > 1.$	$\mathcal{O}((q_1 + 1)n^{\frac{3q_1-2q_2+1}{2(q_1-q_2)}} \sqrt{n} \log \frac{n}{\epsilon}),$ $q_1 > q_2 > 1.$	[2] Achache (2015)
13	$\frac{t^2-1}{2} + \frac{q^{\frac{1}{t}} - 1}{\log q}, t > 0$	$\mathcal{O}(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}),$ $q=1+\mathcal{O}(n).$	–	[3] Achache (2016)
14	$p \frac{t^2-1}{2} + e^{p(\frac{1}{t}-1)} - 1, p > 0$	$\begin{cases} \mathcal{O}(\sqrt{np^5} \log^2 np \log \frac{n}{\epsilon}) & \text{si } p \geq 1 \\ \mathcal{O}(\sqrt{np^{-5}} \log^2 np \log \frac{n}{\epsilon}) & \text{si } p \leq 1 \end{cases}$	$\begin{cases} \mathcal{O}(\sqrt{np^5} \log \frac{n}{\epsilon}) & \text{si } p \geq 1 \\ \mathcal{O}(\sqrt{np^{-5}} \log \frac{n}{\epsilon}) & \text{si } p \leq 1 \end{cases}$	[17] Bouafia et al.(2016)
15	$\frac{t^2-1}{2} + \frac{4}{p\pi} \left(\tan^p \left(\frac{\pi}{2+2x} \right) - 1 \right)$ $p \geq 2$	$\mathcal{O}(pn^{\frac{p+2}{2(p+1)}} \log \frac{n}{\epsilon})$	$\mathcal{O}(p^2 \sqrt{n} \log \frac{n}{\epsilon})$	[18] Bouafia et al.(2016)
16	$\frac{t^2-1}{2} + \frac{q^{\frac{1}{t}-1} - 1}{q \log q} - \frac{q-1}{q}(t-1), t > 0$	$\mathcal{O}(\sqrt{n} \log n \log \frac{n}{\epsilon})$ si $q=1+\mathcal{O}(n)$	–	[6] Achache et al.(2018)
17	$t^2 + \frac{t^{1-q}}{q-1} - \frac{q}{q-1} + \frac{4}{\pi p} [\tan^p h(t) - 1],$ $\begin{cases} h(t) = \frac{\pi}{2t+2}, \\ p \geq 2, q > 1. \end{cases}$	$\mathcal{O}(pn^{\frac{(p+2)(q+1)}{2(p+1)q}} \log \frac{n}{\epsilon})$	$\mathcal{O}((p+q)^2 \sqrt{n} \log \frac{n}{\epsilon})$	[20] Bouafia et al.(2019)

Dans notre travail, on propose une nouvelle fonctions noyaux basées sur un terme barrière double : c'est une combinaison de la fonction barrière classique et une fonction exponentielle.

$$\psi(t) = \frac{t^2 - 1 - \log t}{2} + \frac{e^{\frac{1}{t^q} - 1} - 1}{2q} \quad \text{pour } t > 0, q \geq 1..$$

Théoriquement, on démontre que l'algorithme possède une complexité polynômiale en donnant la complexité des méthodes à grand et à petit pas. Numériquement, on propose plusieurs choix du pas de déplacement afin d'améliorer le comportement numérique de notre algorithme. Cette étude est réalisée sur deux classes de problèmes à savoir : La programmation linéaire et la programmation semi-définie.

Cette thèse est répartie en trois chapitres. Le premier chapitre présente un rappel des notions fondamentales d'usage fréquent, à savoir : l'analyse matricielle, l'analyse convexe, la notion des cônes des matrices symétriques semi-définies positives, en donnant leurs principales propriétés et quelques résultats de la programmation mathématique, servant par la suite aux deux autres chapitres. Le deuxième chapitre représente une synthèse théorique et une étude algorithmique pour la résolution d'un programme linéaire et semi-défini par les méthodes de points intérieurs basées sur la méthode de trajectoire centrale. Enfin, le troisième chapitre contient deux parties : dans la première partie, on développe un algorithme de points intérieurs de type primal-dual basé sur une nouvelle fonction noyau. Pour analyser la complexité de cet algorithme, on suit le schéma présenté par Bai [12]. Puis on présente des résultats numériques issus de l'application de cet algorithme aux problèmes de la programmation linéaire. Dans la deuxième partie, on généralise l'étude de la première partie au cas semi-défini. On obtient la même complexité qu'au cas linéaire c'est-à-dire

$$\mathcal{O}(q\sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log \frac{n}{\epsilon})$$

pour les méthodes à grand pas et

$$\mathcal{O}(q^{\frac{3}{2}}(\log \sqrt{q})^{\frac{q+1}{q}} \sqrt{n} \log \frac{n}{\epsilon})$$

pour les méthodes à petit pas.

De plus, on effectue quelques tests numériques pour montrer l'efficacité de cette approche et de comparer nos résultats à ceux obtenus par les méthodes basées sur différentes fonctions noyaux. Enfin, nous clôturons ce travail par une conclusion et perspectives.

Notations

\mathbb{R}^n	:	L'espace des vecteurs réels de dimension n .
\mathbb{R}_+	=	$[0, +\infty[$
\mathbb{R}_+^n	:	L'orthant positif de l'espace \mathbb{R}^n .
$\mathbb{R}^{n \times n}$:	L'espace des matrices carrées d'ordre n .
x^T	:	Le vecteur transposé de x .
x_i	:	La $i^{\text{ème}}$ composante de x .
x^k	:	Le $k^{\text{ième}}$ vecteur d'une suite de vecteurs.
$x \geq 0$ ($x > 0$)	=	$x_i \geq 0$ ($x_i > 0$), $\forall i$.
xz	=	$(x_1 z_1, \dots, x_n z_n)^T$ (produit d' <i>Hadamard</i>).
$\frac{x}{z}$	=	$(\frac{x_1}{z_1}, \dots, \frac{x_n}{z_n})^T$ ($z_i \neq 0$, $i = 1, \dots, n$).
\sqrt{x}	=	$(\sqrt{x_1}, \dots, \sqrt{x_n})^T$, ($x \geq 0$).
e	=	$(1, \dots, 1)^T \in \mathbb{R}^n$.
$diag(x)$	=	X (La matrice diagonale avec $X_{ii} = x_i$).
v	=	$\sqrt{\frac{xz}{\mu}}$.
I	:	Matrice identité d'ordre n .
$0_{m \times n}$:	La matrice nulle de type (m, n) .
A^T	:	La matrice transposée de A .
$A \succeq 0$ ($A \succ 0$)	:	A est une matrice semi-définie positive (définie positive).
S_n	=	$\{X : X \in \mathbb{R}^{n \times n}, X^T = X\}$.
S_n^+	=	$\{X : X \in S_n, X \succeq 0\}$.
S_n^{++}	=	$\{X : X \in S_n, X \succ 0\}$.
X^k	:	Le $k^{\text{ième}}$ terme d'une suite de matrices.
A^{-1}	:	L'inverse d'une matrice régulière A .
$\lambda_i(A)$:	La $i^{\text{ème}}$ valeur propre d'une matrice $A \in \mathbb{R}^{n \times n}$.

$$\begin{aligned}
\text{tr}(A) &= \sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i(A), \text{ (la trace d'une matrice } A \in \mathbb{R}^{n \times n} \text{)}. \\
\rho(A) &= \max |\lambda_i(A)|, \text{ (le rayon spectral de } A \text{)}. \\
\|A\|_2 &= \sqrt{\rho(A^T A)}, \text{ (la norme spectral de } A \text{)}. \\
\|A\|_F &= \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_i \sum_j a_{ij}^2}. \\
A \bullet B &= \text{tr}(A^T B), \forall A, B \in \mathbb{R}^{n \times n}. \\
A \sim B &: \text{ Il existe une matrice inversible } P \text{ telle que } A = PBP^{-1} \text{ (semblable)}. \\
\nabla f(x) &= \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T, \text{ (le gradient de } f : \mathbb{R}^n \rightarrow \mathbb{R} \text{)}. \\
\nabla^2 f(x) &= \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{1 \leq i, j \leq n}, \text{ (la matrice hessienne de } f \text{)}.
\end{aligned}$$

Terminologie

- (*PM*) : Programme mathématique.
- (*DPM*) : Le dual de (*PM*).
- (*PL*) : Programme linéaire.
- (*TC*) : La trajectoire centrale.
- (*IPM*) : Méthode de points intérieurs.
- (*CPI*) : Condition de points intérieurs.
- (*SDP*) : Programmation linéaire semi-définie.
- (*DSDP*) : Le dual de (*SDP*).
- (*KKT*) : Karush-Kuhn-Tucker.
- \mathcal{C}^k : Espace des fonctions k fois continûment différentiables.

Chapitre 1

Rappels de base

Dans ce chapitre, nous allons introduire certaines notions et résultats bien connus sur les matrices symétriques et les matrices semi-définies positives, ainsi que des notions de base de l'analyse convexe et de la programmation linéaire. Ces notions sont utiles pour démontrer des résultats théoriques dans les chapitres suivants.

1.1 Analyse convexe

La notion de convexité est un outil mathématique important pour l'étude théorique et numérique des problèmes d'optimisation. A ce propos, nous présentons dans ce paragraphe quelques notions de base d'usage courant.

Définition 1.1.1 *Un sous-ensemble $C \subseteq \mathbb{R}^n$ est dit convexe si :*

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

Définition 1.1.2 *Soient x_1, \dots, x_k des points de \mathbb{R}^n , on dit que $x \in \mathbb{R}^n$ est une combinaison convexe de x_1, \dots, x_k , s'il existe $(\mu_1, \dots, \mu_k) \in \mathbb{R}_+^k$ tels que, $x = \sum_{i=1}^k \mu_i x_i$ avec*

$$\sum_{i=1}^k \mu_i = 1.$$

Définition 1.1.3 Etant donné $\mathbf{C} \subset \mathbb{R}^n$, l'enveloppe convexe de \mathbf{C} , notée $\text{Conv}(\mathbf{C})$, est définie par : $\text{Conv}(\mathbf{C}) =$ ensemble de toutes les combinaisons convexes d'éléments de \mathbf{C} .

Définition 1.1.4 L'intersection d'un nombre quelconque d'ensembles convexes est convexe.

Définition 1.1.5 On dit que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ définie sur un ensemble convexe $\mathbf{C} \subset \mathbb{R}^n$ est convexe si : $\forall x, y \in \mathbf{C}, \forall \lambda \in [0, 1]$, on a :

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y),$$

ou d'une manière équivalente si : $\forall k \in \mathbb{N}^*, \forall \mu_i \in \mathbb{R}^+, \sum_{i=1}^k \mu_i = 1$ et $x_i \in \mathbf{C}$, on a :

$$f\left(\sum_{i=1}^k \mu_i x_i\right) \leq \sum_{i=1}^k \mu_i f(x_i),$$

f est strictement convexe si : $\forall x, y \in \mathbf{C}, x \neq y$ et $0 < \lambda < 1$, on a :

$$f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y).$$

Théorème 1.1.1 Une combinaison linéaire à coefficients positifs des fonctions convexes est une fonction convexe.

Théorème 1.1.2 Si $f \in \mathcal{C}^2(\Omega)$, Ω ouvert et $\mathbf{C} \subset \Omega$ convexe, les conditions suivantes sont équivalentes :

1- f est convexe sur \mathbf{C} .

2- $\forall x, y \in \mathbf{C}, f(y) \geq f(x) + (y - x)^T \nabla f(x)$.

3- $\forall x \in \mathbf{C}$, la matrice $H(x) = \nabla^2 f(x)$ (le Hessien de f) est semi-définie positive dans \mathbf{C} .

Définition 1.1.6 Une fonction f est dite coercive sur un ensemble convexe \mathbf{C} si :

$$\lim_{\substack{\|x\| \rightarrow +\infty \\ x \in \mathbf{C}}} f(x) = +\infty.$$

1.2 Analyse matricielle

On notera $\mathbb{R}^{m \times n}$ l'espace vectoriel des matrices réelles $m \times n$. L'espace vectoriel des matrices carrées symétriques d'ordre n est noté S_n . La trace d'une matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ est une fonction linéaire définie par $tr(A) = \sum_{i=1}^n a_{ii}$. Il est important de signaler qu'elle est égale à la somme des valeurs propres de A . Dans cet espace vectoriel le produit scalaire, entre deux éléments $A = (a_{ij}), B = (b_{i,j})$ est défini par :

$$A \bullet B = tr(B^T A) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij} = B \bullet A.$$

La norme associée à ce produit scalaire est définie par :

$$\|A\| = \sqrt{A \bullet A} = \sqrt{tr(A^T A)} \text{ pour tout } A \in \mathbb{R}^{n \times n}.$$

A la place de $\mathbb{R}^{m \times n}$, on travaillera habituellement dans S_n qui est isomorphe à $\mathbb{R}^{\frac{n(n+1)}{2}}$. Pour $A, B \in S_n$, on a ;

$$A \bullet B = tr(BA) \text{ et } \|A\| = \sqrt{\sum_{j=1}^n \lambda_j^2(A)}.$$

Toutes les valeurs propres d'une matrice $A \in S_n$ sont réelles et en plus il existe une matrice orthogonale $P \in \mathbb{R}^{n \times n}$ qui diagonalise A , c'est à dire $P^T A P = \Lambda_A$ où Λ_A est une matrice diagonale dont les éléments diagonaux sont les valeurs propres de A . Dans nos besoins, il convient d'écrire les valeurs propres dans l'ordre décroissant :

$$\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_n(A) = \lambda_{\min}(A).$$

Pour une matrice $A \in S_n$ avec $\text{rang}(A) = k, k \leq n$, la décomposition spectrale de A , $A = P \Lambda_A P^T$, est donnée par une matrice diagonale $\Lambda_A \in S_k$ dont les éléments diagonaux sont les valeurs propres non nulles de A sur sa diagonale principale, et une matrice

$P \in \mathbb{R}^{n \times k}$ telle que $P^T P = I_k$.

Définition 1.2.1 $A \in S_n$ est semi-définie positive ($A \in S_n^+$ ou $A \succeq 0$) si $x^T A x \geq 0$ $\forall x \in \mathbb{R}^n$. $A \in S_n$ est définie positive ($A \in S_n^{++}$ ou $A \succ 0$) si $x^T A x > 0$, $\forall x \in \mathbb{R}^n \setminus \{0\}$.

On énonce certaines conséquences immédiates de ces définitions qui seront utilisées par la suite.

Remarque 1.2.1 Toute sous-matrice principale d'une matrice semi-définie (resp. définie) positive est aussi semi-définie (resp. définie) positive. En particulier, tous les éléments diagonaux d'une matrice semi-définie (resp. définie) positive doivent être positifs (resp. strictement positifs).

Remarque 1.2.2 Pour $A \in S_n^+$, il existe $i \in \{1, \dots, n\}$:

$$a_{ii} = \max \{|a_{ij}| : i, j \in \{1, \dots, n\}\}.$$

Remarque 1.2.3 Si $A \in S_n^+$ et $a_{ii} = 0$ pour certain $i \in \{1, \dots, n\}$ alors $a_{ij} = 0$ pour tout $j \in \{1, \dots, n\}$.

Proposition 1 Soit $B \in \mathbb{R}^{n \times n}$ une matrice régulière. Alors $A \in S_n^+$ si et seulement si $B^T A B \in S_n^+$ et $A \in S_n^{++}$ si et seulement si $B^T A B \in S_n^{++}$.

Lemme 1.2.1 [46] Soient $A \in S_n^+$ et $B \in S_n^+$, alors, les propriétés suivantes :

1. $A \succeq 0$, $B \succeq 0$ et $Tr(AB) = 0$,
2. $A \succeq 0$, $B \succeq 0$ et $\left\|A^{\frac{1}{2}}B^{\frac{1}{2}}\right\|_F^2 = 0$,
3. $A \succeq 0$, $B \succeq 0$ et $AB = 0$.

sont équivalentes.

Théorème 1.2.1 (Théorème de décomposition spectrale d'une matrice symétrique)

Soit $X \in S_n$ et

$$X = Q_X^{-1} \text{diag}(\lambda_1(X), \lambda_2(X), \dots, \lambda_n(X)) Q_X,$$

la décomposition spectrale de X , où $\lambda_i(X)$, $1 \leq i \leq n$, sont les valeurs propres de X et Q_X est une matrice orthogonale ($Q_X^T Q_X = I$). La fonction matricielle ψ est définie par :

$$\psi(X) = Q_X^{-1} \text{diag}(\psi(\lambda_1(X)), \dots, \psi(\lambda_n(X))) Q_X,$$

où ψ est une fonction différentiable à valeurs réelles.

On note que la fonction ψ' est bien définie lorsque $\psi'(t)$ est bien définie à chaque valeur propre de X et $\psi'(X) = Q_X^{-1} \text{diag}(\psi'(\lambda_1(X)), \dots, \psi'(\lambda_n(X))) Q_X$.

Ce théorème nous permet de faire l'extension de la définition d'une fonction $\psi : \mathbb{R} \rightarrow \mathbb{R}$ à une fonction de S_n dans S_n (voir [34]). Maintenant, nous avons besoin de quelques concepts liés à la fonction matricielle. Pour plus de détails, consulter les ouvrages ([34], [45]).

Définition 1.2.2 Une matrice $X(t)$ est dite matrice de fonction si chaque élément de $X(t)$ est une fonction de t c-à-d $X(t) = [X_{ij}(t)]$.

Les concepts habituels de continuité, de différentiabilité, d'intégrabilité et certaines règles de base du calcul peuvent être étendus aux fonctions à valeurs matricielles. Alors, on a :

$$\left\{ \begin{array}{l} X'(t) \\ \frac{d}{dt} \text{tr}(X(t)) \\ \frac{d}{dt} \text{tr}(\psi(X(t))) \\ \frac{d}{dt}(X(t)G(t)) \end{array} \right. = \begin{array}{l} \frac{d}{dt} X(t) = \left(\frac{d}{dt} X_{ij}(t) \right) \\ \text{tr} \left(\frac{d}{dt} X(t) \right) = \text{tr}(X'(t)) \\ \text{tr}(\psi'(X(t))) X'(t) \\ \left(\frac{d}{dt} X(t) \right) G(t) + X(t) \left(\frac{d}{dt} G(t) \right) \\ X'(t) G(t) + X(t) G'(t) \end{array}$$

Théorème 1.2.2 (Caractérisation des matrices définies positives) Pour $A \in S_n$, les propriétés suivantes :

1. $A \in S_n^{++}$,
2. $\lambda_i(A) > 0 \quad \forall i = 1, \dots, n$,
3. Il existe $C \in \mathbb{R}^{n \times n}$ avec $\text{rang}(C) = n$ tel que $A = C^T C$,

4. Pour une suite arbitraire $A_i \in S_i, i = 1, \dots, n$, de sous-matrices principales de $A : \det(A_i) > 0$ pour $i = 1, \dots, n$,

5. $A^{-1} \in S_n^{++}$,

sont équivalentes.

Théorème 1.2.3 (Factorisation de Cholesky) Pour toute matrice $A \succ 0$, il y a une seule matrice triangulaire inférieure et inversible L telle que $A = L^T L$.

Définition 1.2.3 Une matrice $A \in \mathbb{R}^{n \times n}$ est à diagonale strictement dominante si :

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, \forall i = 1, \dots, n.$$

Théorème 1.2.4 [55] Si $A \in S_n$, est à diagonale strictement dominante et si tous les éléments diagonaux sont strictement positifs, alors A est définie positive.

Théorème 1.2.5 [55] Pour $A \in S_n$, les propriétés suivantes :

1. $A \in S_n^+$.

2. $\lambda_i(A) \geq 0 \quad \forall i = 1, \dots, n$,

3. Il existe $C \in \mathbb{R}^{m \times n}$ telle que $A = C^T C$ avec $\text{rang}(C) = \text{rang}(A)$,

sont équivalentes.

Proposition 2 Soit $A \in S_n^{++}$, alors il existe une matrice unique $B \in S_n^{++}$ telle que $A = B^2$, et on la note souvent par $B = A^{\frac{1}{2}}$. De plus, B est appelée la racine carrée de A .

Cône des matrices symétriques semi-définies positives

Nous allons maintenant considérer le cône des matrices symétriques semi-définies positives comme un sous-ensemble de S_n .

Définition 1.2.4 Un sous ensemble $\mathbb{K} \subset \mathbb{R}^n$ est un cône, s'il est stable pour la multiplication par des scalaires positifs : $\lambda x \in \mathbb{K}, \forall x \in \mathbb{K}, \forall \lambda \geq 0$. Un cône \mathbb{K} est pointu si $\mathbb{K} \cap (-\mathbb{K}) = \{0\}$. De plus, si \mathbb{K} est convexe, alors \mathbb{K} est un cône convexe.

Proposition 3 S_n^+ est un cône pointu et fermé dans $\mathbb{R}^{\frac{n(n+1)}{2}}$. L'ensemble des matrices définies positives S_n^{++} n'est pas un cône car $0 \notin S_n^{++}$.

Définition 1.2.5 Un cône \mathbb{K} est dit propre si :

1. \mathbb{K} est un cône convexe.
2. \mathbb{K} est fermé (topologique).
3. \mathbb{K} est d'intérieur non vide.
4. \mathbb{K} est pointu.

Lemme 1.2.2 [56] Soient $A, B \in S_n^+$. Alors $A \bullet B \geq 0$ et en plus $A \bullet B = 0$ si et seulement si $AB = 0$.

Corollaire 1.2.1 (Théorème de trace de Fejer) $A \in S_n^+$ si et seulement si

$$A \bullet B \geq 0, \forall B \in S_n^+.$$

Le cône des matrices symétriques semi-définies positives induit une relation d'ordre partiel sur l'ensemble des matrices symétriques, définie par

$$A \succeq B \Leftrightarrow (A - B) \in S_n^+.$$

1.3 Programmation mathématique

D'une façon générale, un programme mathématique est un problème d'optimisation avec contraintes de type :

$$(PM) \min_{x \in \mathbf{D}} f(x) \quad \text{où} \quad \mathbf{D} = \{x \in \mathbb{R}^n / g_i(x) \leq 0, i = 1, \dots, m \text{ et } h_j(x) = 0, j = 1, \dots, p \}$$

et f, g_i, h_j sont des fonctions données de \mathbb{R}^n dans \mathbb{R} . On appelle f la fonction objectif et \mathbf{D} l'ensemble des solutions réalisables ou ensemble des contraintes ou tout simplement "le

domaine". La classification de (PM) et son traitement numérique sont établis à partir des propriétés fondamentales des fonctions f, g_i, h_j à savoir la convexité, la différentiabilité et la linéarité.

On appelle fonction barrière de (PM) toute fonction F telle que :

- * F est à valeurs finies sur l'intérieur (relatif) du domaine réalisable \mathbf{D} ,
- * $F(x) \rightarrow \infty$ quand $x \rightarrow \text{Fr}(\mathbf{D})$ (frontière de \mathbf{D}).

1.3.1 Existence et unicité de solution

Dans ce paragraphe, nous donnons les théorèmes d'existence et d'unicité les plus utilisés (voir [16, 39]) :

Théorème 1.3.1 (Weirstrass) *Si \mathbf{D} est un fermé borné (compact) non vide de \mathbb{R}^n et si f est continue sur \mathbf{D} alors (PM) admet au moins une solution optimale $x^* \in \mathbf{D}$.*

Corollaire 1.3.1 *Si \mathbf{D} est fermé non vide de \mathbb{R}^n , f est continue et coercive (c-à-d $f(x) \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$) sur \mathbf{D} , alors (PM) admet au moins une solution optimale.*

Théorème 1.3.2 *Si \mathbf{D} est convexe non vide de \mathbb{R}^n , f est strictement convexe sur \mathbf{D} alors (PM) admet au plus une solution optimale.*

1.3.2 Conditions d'optimalité

Avant de donner les conditions d'optimalité de (PM) , on exige que les contraintes doivent satisfaire certains critères dits "critères de qualification".

- Une contrainte g_i est dite active (ou saturée) en $\bar{x} \in \mathbf{D}$ si $g_i(\bar{x}) = 0$.
- Un point $\bar{x} \in \mathbf{D}$ est dit régulier (on dit également que les contraintes sont qualifiées en \bar{x}) si les composantes du gradient, correspondant aux contraintes saturées en \bar{x} , sont linéairement indépendantes.

Il existe aussi deux critères usuels de qualification en tout point de \mathbf{D} , à savoir :

- Si toutes les contraintes sont affines.
- Si D est défini uniquement par des inégalités.

On a le critère de Slater suivant : g_i est convexe pour tout $i = 1, \dots, m$ et qu'il existe un point x^0 telle que : $g_i(x^0) < 0, (int(D) \neq \emptyset)$.

Théorème 1.3.3 (Karush - Kuhn - Tucker) Soit $\bar{x} \in D$ satisfaisant l'une des conditions de qualification et supposons que f, g_i, h_j sont $C^1(\mathbb{R}^n)$, on a :

Si \bar{x} est un optimum local pour (PM), alors il existe des réels $\lambda_i \in \mathbb{R}^+, i = 1, \dots, m$ et $\gamma_j \in \mathbb{R}, j = 1, \dots, p$, dits multiplicateurs de Lagrange. Telles que :

$$\left\{ \begin{array}{l} \nabla f(\bar{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\bar{x}) + \sum_{j=1}^p \gamma_j \nabla h_j(\bar{x}) = 0, \quad (\text{conditions d'optimalité}) \\ \lambda_i g_i(\bar{x}) = 0, \quad i = 1, \dots, m, \quad (\text{conditions de complémentarité}) \\ h_j(\bar{x}) = 0, \quad j = 1, \dots, p. \end{array} \right.$$

Si de plus, f, g_i, h_j sont convexes, les conditions précédentes sont à la fois nécessaires et suffisantes pour que \bar{x} soit un optimum global pour (PM).

1.3.3 Dualité d'un programme mathématique

Considérons le problème (PM), on associe à ce problème la fonction lagrangienne suivante :

$$L(x, \lambda, \gamma) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \gamma_j h_j(x), x \in \mathbb{R}^n, \quad \lambda_i \geq 0, \gamma_j \in \mathbb{R}.$$

Le dual de (PM) au sens de Lagrange est donné par :

$$(DPM) \quad \max_{\lambda \geq 0, \gamma \in \mathbb{R}^p} \min_{x \in \mathbb{R}^n} L(x, \lambda, \gamma) = \max_{\lambda \geq 0, \gamma \in \mathbb{R}^p} q(\lambda, \gamma).$$

Sachant que q désigne la fonction duale de f et vérifie :

$$q(\lambda, \gamma) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \gamma), \forall \lambda \in \mathbb{R}_+^n \text{ et } \gamma \in \mathbb{R}^p.$$

1.3.4 Algorithme d'optimisation

Un algorithme est défini par une application $\mathcal{A} : \mathbf{D} \rightarrow \mathbf{D}$, où \mathbf{D} est l'ensemble des solutions réalisables, permettant la génération d'une suite d'éléments de \mathbf{D} par la formule :

$$\begin{cases} x_0 \in \mathbf{D} \text{ donné, } k = 0 & \text{Etape d'initialisation} \\ x_{k+1} = \mathcal{A}(x_k), \quad k = k + 1 & \text{Itération} \end{cases}.$$

Si on remplace \mathbf{D} par son intérieur, en supposant que $\text{int}(\mathbf{D}) \neq \emptyset$, l'algorithme est dit un algorithme de points intérieurs. Définir un algorithme, n'est autre que construire une suite $(x_k)_{k \in \mathbb{N}}$ de \mathbf{D} et réaliser une étude pour montrer sa convergence.

Définition 1.3.1 *On dit que l'algorithme \mathcal{A} est convergent si la suite $(x_k)_{k \in \mathbb{N}}$ engendrée par l'algorithme converge vers une limite x^* .*

Taux de convergence

Un critère de mesure de la vitesse (ou le taux) de convergence est l'évolution de l'erreur commise à chaque itération ($e_k = \|x_k - x^*\|$).

Avant de donner les notations de convergence, nous donnons les définitions des notations asymptotiques suivantes :

Définition 1.3.2 *Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$. On note*

$$* f(n) = \mathcal{O}(g(n)) \Leftrightarrow \exists k > 0, \exists n_0, \forall n > n_0 \quad |f(n)| \leq k |g(n)|.$$

$$* f(n) = \Theta(g(n)) \Leftrightarrow \exists k_1, k_2 > 0, \exists n_0, \forall n > n_0 \quad k_1 |g(n)| \leq |f(n)| \leq k_2 |g(n)|.$$

La classification de la vitesse de convergence d'une suite est basée sur les notions de

comparaison des fonctions au voisinage de $+\infty$. La vitesse de la convergence pourra être :

1. **Linéaire** : S'il existe une norme $\|\cdot\|$, un réel $\tau \in [0, 1[$ et un indice k_1 , tels que :

$$\forall k \geq k_1, \|x_{k+1} - x^*\| \leq \tau \|x_k - x^*\|.$$

2. **Superlinéaire** :

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0.$$

3. **D'ordre γ ($\gamma > 1$)** : S'il existe une constante C telle que :

$$\forall k, \|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^\gamma.$$

1.3.5 Programmation linéaire

Un programme linéaire (*PL*) est un problème d'optimisation qui consiste à maximiser (ou minimiser) une fonction linéaire sous contraintes linéaires, il s'agit donc d'un programme mathématique, où la fonction objectif est linéaire et l'ensemble des contraintes est affine.

Ecriture mathématique : Il existe trois formes pour écrire un programme linéaire :

La forme canonique :

$$\left\{ \begin{array}{l} \min_x c^T x, 0 \neq c \in \mathbb{R}^n \text{ (vecteur donné)} \\ Ax \geq b, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \text{ (donnés)} \\ x \geq 0, x \in \mathbb{R}^n \text{ (variable)}. \end{array} \right.$$

La forme standard :

$$\left\{ \min_x c^T x, Ax = b, x \geq 0. \right.$$

La forme générale :

$$\left\{ \min_x c^T x, Ax \leq b, Bx \geq b', B \in \mathbb{R}^{p \times n}, b' \in \mathbb{R}^p. \right.$$

1.3.6 Dualité en programmation linéaire

On considère un programme linéaire sous la forme standard :

$$(P) \quad \min_x c^T x : Ax = b, x \geq 0,$$

où $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

Bien évidemment, n'importe quel (PL) se ramène facilement à cette forme.

Le dual de (P) est défini par :

$$(D) \quad \max_{(y, z)} b^T y : A^T y + z = c, y \in \mathbb{R}^m, z \in \mathbb{R}_+^n.$$

On notera par la suite :

$$\mathcal{F}_{(P)} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

$$\mathcal{F}_{(P)}^0 = \{x \in \mathbb{R}^n : Ax = b, x > 0\}$$

et

$$\mathcal{F}_{(D)} = \{(y, z) \in \mathbb{R}^{m \times n} : A^T y + z = c, z \geq 0\}$$

$$\mathcal{F}_{(D)}^0 = \{(y, z) \in \mathbb{R}^{m \times n} : A^T y + z = c, z > 0\},$$

les ensembles des solutions réalisables et strictement réalisables des deux problèmes (P) et (D), respectivement.

On suppose que les problèmes (P) et (D) vérifient les hypothèses suivantes :

Hypothèse 1 : La matrice A est de plein rang. ($\text{rang } A = m < n$).

Hypothèse 2 : Condition de points intérieurs (CPI), il existe (x^0, y^0, z^0) tel que :

$$Ax^0 = b, A^T y^0 + z^0 = c, x^0 > 0, z^0 > 0, \text{ c.à.d. } \mathcal{F}_{(P)}^0 \times \mathcal{F}_{(D)}^0 \neq \emptyset$$

Exemple 1

$$(P) \begin{cases} \min(x_1 + x_2) \\ x_1 - x_2 = 0 \\ x_1 + x_2 + x_3 = 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{cases}$$

Le programme dual est :

$$(D) \begin{cases} \max y_2 \\ y_1 + y_2 \leq 1 \\ -y_1 + y_2 \leq 1 \\ y_2 \leq 0. \end{cases}$$

Théorème 1.3.4 (Dualité faible) Soient (P) et (D) deux (PL) alors :

$$c^T x \geq b^T y, \quad \forall x \in \mathcal{F}_{(P)}, \forall (y, z) \in \mathcal{F}_{(D)}.$$

Théorème 1.3.5 (Dualité forte) Si $x \in \mathcal{F}_{(P)}$ et $(y, z) \in \mathcal{F}_{(D)}$, tels que

$$c^T x = b^T y,$$

alors, x et (y, z) sont des solutions optimales de (P) et (D) , respectivement.

Théorème 1.3.6 (Théorème des écarts complémentaires) Soient x et (y, z) deux solutions réalisables primale et duale, respectivement et $z = c - A^T y$, le vecteur des variables d'écarts associé à y , alors x et (y, z) sont optimales si :

$$xz = 0, (x_i z_i = 0, \forall i = 1, \dots, n).$$

Définition 1.3.3 Une base de A est une sous matrice B formée de m colonnes linéairement indépendantes de A .

• La solution de base associée à B est le point $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} \in \mathbb{R}^n$ tel que :

$$x_B = B^{-1}b, x_N = 0.$$

- Une solution de base qui vérifie $x_B \geq 0$ est dite solution de base réalisable.
- Un point x est un sommet de $\mathcal{F}_{(P)}$ si et seulement si il est une solution de base réalisable.
- Un sommet est dit non dégénéré si $x_B > 0$. Il est dégénéré dans le cas contraire (au moins une composante de x_B est nulle).

Théorème 1.3.7 *Si un programme linéaire possède une solution optimale, alors au moins un sommet du domaine réalisable est une solution optimale.*

Chapitre 2

Méthodes de points intérieurs

Ces méthodes ont été développées dans les années 60 dans le but de résoudre des programmes mathématiques non linéaires. Leurs utilisations pour la programmation linéaire n'ont pas reçu autant d'enthousiasme à cause de la dominance quasi totale de la méthode du simplexe à cette époque. Après l'apparition de l'algorithme de Karmarkar [36] en 1984 pour la programmation linéaire, les méthodes de points intérieurs ont connu une véritable révolution. Depuis, plusieurs publications dans ce domaine ont été faites. On distingue trois classes fondamentales de méthodes de points intérieurs à savoir : les méthodes affines, les méthodes de réduction du potentiel et les méthodes de trajectoire centrale (suivi du chemin) (en anglais : Path-following methods).

a/ Méthodes affines (Dikin 1967) : Il s'agit pratiquement de l'algorithme de Karmarkar sans fonction potentielle et sans transformation projective, on utilise une transformation affine et on remplace la contrainte de non négativité par un ellipsoïde qui contient le nouvel itéré. L'algorithme est d'une structure simple, malheureusement, il n'est pas facile de démontrer la polynômialité.

b/ Méthodes de réduction du potentiel : La fonction potentielle joue un grand rôle dans le développement des méthodes de points intérieurs. L'algorithme de Karmarkar appliqué au programme linéaire sous forme standard utilise une fonction potentielle de

la forme :

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

où $\rho = n + 1$ et Z est une borne inférieure de la valeur optimale de l'objectif. Karmarkar [36] prouve la convergence et la polynômialité de son algorithme en montrant que cette fonction est réduite à chaque itération par au moins une constante. Depuis 1987, les chercheurs introduisent des fonctions potentielles de type primal-dual, parmi lesquelles, celle de Tanabe, Todd [52] et Ye [56] définie par :

$$\Phi_\rho(x, z) = \rho \log(x^T z) - \sum_{i=1}^n \log x_i z_i, \text{ pour } \rho > n.$$

Cette fonction a joué un rôle très important dans le développement des algorithmes de réduction du potentiel après 1988. Les algorithmes correspondants à ces méthodes possèdent une complexité polynômiale, ils nécessitent $\mathcal{O}(\sqrt{n} |\log \epsilon|)$ itérations pour réduire le saut de dualité ($x^T z \leq \epsilon$, ϵ est une précision donnée).

c/ Méthodes de trajectoire centrale (TC) : Elles ont été introduites à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles possèdent de bonnes propriétés théoriques : une complexité polynômiale et une convergence super linéaire. Les algorithmes de trajectoire centrale restreignent les itérés à un voisinage de la trajectoire centrale, ce dernier est une courbe de points strictement réalisables. (voir Wright [56] et T. Terlaky [51]). Notre travail se base sur une méthode de trajectoire centrale pour la programmation linéaire et l'extension à la programmation semi-définie (*SDP*). Pour cela, on donne tout d'abord, un aperçu sur la méthode (*TC*) en programmation linéaire et semi-définie.

2.1 Méthodes de trajectoire centrale (TC) pour la programmation linéaire

L'idée générale de cette méthode consiste à suivre un chemin particulier (trajectoire centrale) constitué d'une suite de points intérieurs en prenant comme direction de déplacement celle de Newton, et un pas de déplacement convenable choisi d'une façon à maintenir la stricte faisabilité du nouvel itéré. Les méthodes primales-duales de points intérieurs rentrent dans le cadre du chemin central et sont les méthodes les plus efficaces du point de vue pratique, en particulier, pour les problèmes de grande taille. Actuellement les chercheurs s'intéressent à l'étude de l'amélioration du comportement de l'algorithme en regardant en particulier la complexité algorithmique des méthodes de trajectoire centrale, en utilisant de nouvelles techniques.

2.1.1 Méthodes de trajectoire centrale classique

Présentation de la méthode

On considère le programme linéaire suivant, sous forme standard :

$$\begin{cases} \min_x c^T x \\ Ax = b \\ x \geq 0, \end{cases} \quad (P)$$

où $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ avec $\text{rang}(A) = m < n$.

On associe à (P) le problème perturbé, défini comme suit :

$$\begin{cases} \min_x f_\mu(x) = c^T x - \mu \sum_{i=1}^n \log x_i \\ Ax = b \\ x > 0, \end{cases} \quad (P_\mu)$$

où μ est un paramètre barrière strictement positif.

Lemme 2.1.1 [56] *La fonction f_μ est strictement convexe. De plus si $\mathcal{F}_{(P)}^0$ est non vide, alors pour tout $\mu > 0$, le problème (P_μ) admet une solution unique, notée $x(\mu)$, et appelée point centre. Quand $\mu \rightarrow 0$, $x(\mu) \rightarrow x^*$ (solution optimale de (P)).*

$x(\mu)$ est définie d'une façon unique par les conditions d'optimalité de Karush-Khuhn-Tucker suivantes :

$$\begin{cases} c - \mu X^{-1}e - A^T y = 0 \\ Ax = b, \end{cases} \quad (S)$$

où $y \in \mathbb{R}^m$ est le multiplicateur de Lagrange associé à la contrainte $Ax = b$, du problème (P_μ) . Posons $z = \mu X^{-1}e \in \mathbb{R}_{++}^n$, le système précédent devient :

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

Le système (S_μ) est un système d'équations non linéaires, pour cela, la méthode de Newton est l'une des méthodes les plus utilisées pour sa résolution. A chaque μ , en appliquant la méthode de Newton, on trouve une solution $(x(\mu), y(\mu), z(\mu))$. La fonction $\mu \rightarrow (x(\mu), y(\mu), z(\mu))$ définit la trajectoire centrale qu'on note

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

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

$$T_C(\theta) = \{(x, y, z) \in \mathcal{F}_{(P)}^0 \times \mathcal{F}_{(D)}^0 : \|XZ e - \mu e\| \leq \theta \mu, \quad 0 < \theta < 1\}.$$

Notons que (S_μ) correspond aux conditions de complémentarité pour un programme linéaire primal-dual.

Le système (S_μ) désigne aussi les conditions d'optimalité du problème dual paramétrisé

(D_μ) suivant :

$$\begin{cases} \max b^T y + \mu \sum_{i=1}^n \log z_i, & \mu > 0 \\ A^T y + z = c, & z > 0. \end{cases} \quad (D_\mu)$$

En effet, les conditions d'optimalité de Karush - Kuhn - Tucker pour ce dernier problème sont données par :

$$\begin{cases} b - Ax & = 0 \\ \mu Z^{-1} e - x & = 0 \\ A^T y + z & = c. \end{cases}$$

En calculant une racine de la fonction $F(x, y, z) = 0$, où

$$F(x, y, z) = (XZe - \mu e, Ax - b, A^T y + z - c),$$

l'itéré, généré par la méthode de Newton, est défini par :

$$(x^+, y^+, z^+) = (x, y, z) + \alpha(\Delta x, \Delta y, \Delta z), \quad \alpha > 0,$$

où $(\Delta x, \Delta y, \Delta z)$ est une solution du système linéaire :

$$\nabla F(x, y, z) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F(x, y, z). \quad (SL_\mu)$$

s'écrit sous forme matricielle comme suit :

$$\begin{bmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} XZe - \mu e \\ 0 \\ 0 \end{bmatrix}.$$

En pratique, trouver une solution strictement réalisable initiale qui soit proche de la trajectoire centrale est une difficulté majeure faisant l'objet de recherche. Pour remédier

à cette difficulté, les chercheurs ont introduit le paramètre poids, associé à la fonction barrière.

2.1.2 Méthodes de trajectoire centrale avec poids

La méthode de trajectoire centrale avec poids représente une généralisation de la méthode classique. On associe à (P) le problème perturbé, défini comme suit :

$$\min_x f_{\mu r}(x) = c^T x - \mu \sum_{i=1}^n r_i \log x_i : Ax = b, x > 0, \mu > 0, \quad (P_{\mu r})$$

$r = (r_1, \dots, r_n)^T \in \mathbb{R}_+^n$ est le vecteur des poids associé à la fonction barrière .

La fonction $f_{\mu r}$ est strictement convexe et les contraintes sont affines, alors les conditions de Karush-Kuhn-Tucker correspondantes sont nécessaires et suffisantes et s'écrivent comme suit :

$$\begin{cases} c - \mu X^{-1}r - A^T y = 0, \\ Ax = b, \\ x > 0, y \in \mathbb{R}^m. \end{cases}$$

Posons $z = \mu X^{-1}r \in \mathbb{R}_{++}^n$, le système précédent devient :

$$\begin{cases} A^T y + z = c, \\ Ax = b, \quad x \geq 0 \\ Xz - \mu r = 0, \quad z \geq 0 \quad \text{et } \mu > 0. \end{cases} \quad (S_{\mu r})$$

Pour $r = e$, on trouve le système classique, c'est à dire la méthode primale-duale de trajectoire centrale classique (sans poids). On applique alors la méthode de Newton pour $(x, y, z) \in \mathcal{F}_{(P)}^0 \times \mathcal{F}_{(D)}^0$, on obtient le système :

$$\begin{cases} A\Delta x = 0 \\ A^T \Delta y + \Delta z = 0 \\ X\Delta z + Z\Delta x = -Xz + \mu r, \end{cases} \quad (S'_{\mu r})$$

dont la solution est $(\Delta x, \Delta y, \Delta z)$. Le nouvel itéré est :

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

Le point (x^+, y^+, z^+) qui vérifie le système $(S_{\mu r})$ est voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$T_r(\theta) = \{(x, y, z) \in \mathcal{F}_{(P)}^0 \times \mathcal{F}_{(D)}^0 : \|XZe - \mu r\| \leq \theta\mu, 0 < \theta < 1\}.$$

Calcul du pas de déplacement :

Théoriquement la méthode (*TC*) converge pour tout $0 < \alpha < 1$, mais on sait que le pas de déplacement α et le nombre d'itérations sont inversement proportionnels : " plus α est grand, plus le nombre d'itérations est petit". Le pas de déplacement α est choisi d'une façon à maintenir la stricte faisabilité du nouvel itéré, alors on envisage deux types de méthodes pour calculer un pas convenable.

1)- Procédure de positivité [38] : Pour que le nouvel itéré soit positif, il est nécessaire que

$$\begin{cases} x^+ = x + \alpha\Delta x > 0 & (a) \\ z^+ = z + \alpha\Delta z > 0 & (b) \end{cases}$$

De (a), on a :

$$(\alpha\Delta x > -x) \Leftrightarrow \alpha = \alpha_1 = \begin{cases} \min\left(\frac{-x_i}{(\Delta x)_i}\right) & \text{si } I_0 \neq \emptyset \\ 1 & \text{si } I_0 = \emptyset \end{cases},$$

tel que : $I_0 = \{i : \Delta x_i < 0, i = 1, \dots, n\}$.

De même pour (b) on a :

$$(\alpha\Delta z > -z) \Leftrightarrow \alpha = \alpha_2 = \begin{cases} \min\left(\frac{-z_i}{(\Delta z)_i}\right) & \text{si } I_1 \neq \emptyset \\ 1 & \text{si } I_1 = \emptyset \end{cases},$$

tel que :

$$I_1 = \{i : \Delta z_i < 0, \quad i = 1, \dots, n\}.$$

Alors il suffit de prendre le pas de déplacement suivant :

$$\alpha = \rho \min(\alpha_1, \alpha_2), \quad 0 < \rho < 1.$$

2)- Méthode de recherche linéaire : Le pas de déplacement à chaque itération est la solution du sous problème mono- dimensionnel suivant :

$$(P_\alpha) \begin{cases} \min \varphi(\alpha) = f(x^k + \alpha d_k) \\ \alpha \in \mathbb{R}_+ \text{ (ou } \alpha \in [a, b], a > 0) \end{cases}$$

Le problème (P_α) est appelé problème de recherche linéaire, φ est la fonction de recherche linéaire et f est la fonction objectif du problème. Cette procédure est plus coûteuse en pratique que la procédure de la positivité.

Début algorithme 1

Soit $\epsilon > 0$ (un paramètre de précision), $(x^0, y^0, z^0) \in \mathcal{F}_{(P)}^0 \times \mathcal{F}_{(D)}^0$, $\varrho = \frac{\|X^0 Z^0 e\|}{\sqrt{n}}$,
 $r = \frac{X^0 Z^0 e}{\varrho}$, $\eta = \min \{r_i\}$, $R = \text{diag}(r)$, $\theta = (1 - \frac{\sqrt{2}}{2})\eta$, $\beta = 1 - \frac{\theta}{\sqrt{n}}$ et $\mu^0 = \frac{(x^0)^T z^0}{n}$.

Si $\|X^0(\mu^0)Z^0(\mu^0)e - \mu^0 e\|_2 \leq \mu^0 \theta$, aller à *(TCSP)* (trajectoire centrale sans poids),

Sinon aller à *(TCAP)* (trajectoire centrale avec poids).

Fin si.

(TCSP) : $k = 0$.

Tant que $\mu^k > \epsilon$ faire

Etape 1 : $\mu^{k+1} = \beta \mu^k = \beta \frac{(x^k)^T z^k}{n}$,

Etape 2 : Calculer la direction $(\Delta x^k, \Delta y^k, \Delta z^k)$, en résolvant le système (S'_μ) ,

Etape 3 : Calculer le pas de déplacement α_k ,

Etape 4 : Calculer $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta z^k)$; $k = k + 1$.

Fin tant que.

(TCAP) : $k = 0$.

Tant que $\mu^k > \epsilon$ faire

Etape 1 : $\mu^{k+1} = \beta \mu^k = \beta \frac{(x^k)^T R z^k}{n}$,

Etape 2 : Calculer la direction $(\Delta x^k, \Delta y^k, \Delta z^k)$, en résolvant le système $(S'_{\mu r})$,

Etape 3 : Calculer le pas de déplacement α_k ,

Etape 4 : Calculer $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta z^k)$, $k = k + 1$.

Fin tant que.

Fin.

Algorithme 1 : Méthodes *(TC)* pour *(PL)*

Méthodes de trajectoire centrale non réalisable pour *(PL)*

La méthode de trajectoire centrale non réalisable est une autre alternative qui répond aux questions liées à l'initialisation de démarrer l'algorithme d'un point initial qui vérifie seulement la condition $(x^0, y^0, z^0) > 0$ (non nécessairement réalisable) : En général, la

suite générée par cet algorithme n'est pas réalisable d'où le nom : méthode de trajectoire centrale non réalisable.

Soit $(x^k, y^k, z^k) > 0$, le point obtenu à l'itération k . On applique la méthode de Newton pour déterminer la direction de déplacement $(\Delta x^k, \Delta y^k, \Delta z^k)$, qui est solution du système linéaire suivant :

$$\begin{bmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^t & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} XZe - \mu e \\ Ax - b \\ A^T y + z - c \end{bmatrix}. \quad (S''_{\mu})$$

Le nouvel itéré est donné par :

$$(x^+, y^+, z^+) = (x, y, z) + \alpha(\Delta x, \Delta y, \Delta z), \quad \alpha > 0.$$

Pour réaliser la réalisabilité des itérés et l'optimalité en même temps, on introduit la fonction suivante :

$$\Phi(x, y, z) = \|XZe\|^2 + \|Ax - b\|^2 + \|A^T y + z - c\|^2.$$

L'idée est de faire tendre la fonction Φ vers 0. Le pas de déplacement $\alpha > 0$ est choisi de telle façon que le nouvel itéré $(x^{k+1}, y^{k+1}, z^{k+1})$ reste strictement positif et que Φ décroît à chaque itération, c'est à dire :

$$\Phi(x^{k+1}, y^{k+1}, z^{k+1}) < \Phi(x^k, y^k, z^k).$$

Début algorithme 2

Soit $\epsilon > 0$ (un paramètre de précision), (x^0, y^0, z^0) tel que $(x^0, z^0) > 0$

$k = 0$.

Tant que $\Phi(x^k, y^k, z^k) > \epsilon$ **faire**

Etape1 : $\mu_k = \frac{(x^k)^T z^k}{n}$,

Etape2 : Calculer $(\Delta x^k, \Delta y^k, \Delta z^k)$ solution du système (S''_μ) ,

Etape3 : Trouver un pas de déplacement α_k tel que $(x^k, z^k) > 0$ et

$\Phi(x^{k+1}, y^{k+1}, z^{k+1}) < \Phi(x^k, y^k, z^k)$,

Etape4 : Calculer $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta z^k)$.

$k = k + 1$.

Fin tant que.

Fin.

Algorithme 2 : Méthodes (TC) non réalisable pour (PL)

2.1.3 Méthodes de trajectoire centrale via une fonction noyau

Fonctions auto-régulières et leurs propriétés [46]

Définition 2.1.1 La fonction $\psi(t) \in \mathcal{C}^2 : (0, \infty) \rightarrow \mathbb{R}$ est auto-régulière si elle satisfait les conditions suivantes :

AR1 $\psi(t)$ est strictement convexe pour $t > 0$ et atteint son minimum global au point $t = 1$, telle que :

$$\psi(1) = \psi'(1) = 0.$$

De plus, ils existent des constantes positives $v_2 \geq v_1 > 0$ et $p \geq 1, q \geq 1$, telles que :

$$v_1 (t^{p-1} + t^{-1-q}) \leq \psi''(t) \leq v_2 (t^{p-1} + t^{-1-q}) \quad \forall t \in (0, \infty); \quad (R1)$$

AR2 Pour tous $t_1, t_2 > 0$, on a :

$$\psi(t_1^r t_2^{1-r}) \leq r\psi(t_1) + (1-r)\psi(t_2), \quad r \in [0, 1]. \quad (R2)$$

Nous appelons le paramètre q le degré de barrière et p le degré de croissance de $\psi(t)$, si la fonction ψ est auto-régulière.

Considérons maintenant le cas où $v_2 = v_1 = 1$. Alors

$$(R1) \Rightarrow \psi''(t) = t^{p-1} + t^{-1-q},$$

et, par conséquent, ψ est déterminée uniquement par p et q , elle sera notée $Y_{p,q}(t)$. En intégrant deux fois, on peut facilement vérifier : pour $p \geq 1$,

$$\begin{aligned} Y_{p,1}(t) &= \frac{t^{p+1} - 1}{p(p+1)} - \frac{\log t}{q} + \frac{p-1}{p}(t-1), \\ Y_{p,q}(t) &= \frac{t^{p+1} - 1}{p(p+1)} + \frac{t^{1-q} - 1}{q(q-1)} + \frac{p-q}{pq}(t-1), \quad q > 1. \end{aligned}$$

Puisque

$$\lim_{q \rightarrow 1} \frac{t^{1-q} - 1}{(q-1)} = -\log t,$$

on a

$$\lim_{q \rightarrow 1} Y_{p,q}(t) = Y_{p,1}(t),$$

on peut prolonger par continuité au cas $q = 1$. De plus

$$Y_{1,1}(t) = \frac{t^2 - 1}{2} - \log t$$

Fonctions noyaux et leurs qualifications (éligibilités)[29] :

Définition 2.1.2 Soit $\psi : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$, une fonction deux fois continûment différentiable.

Alors ψ est dite fonction noyau, si elle vérifie les conditions suivantes :

1. $\psi(1) = \psi'(1) = 0$.
2. $\psi''(t) > 0$ pour $t > 0$.
3. $\lim_{t \rightarrow 0^+} \psi(t) = \lim_{t \rightarrow +\infty} \psi(t) = +\infty$.

Les conditions (1) et (2) montrent que ψ est strictement convexe et minimale en 1 avec $\psi(1) = 0$ et implique que ψ s'écrit comme suit

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\tau) d\tau d\xi,$$

tandis que la condition (3) indique que ψ est une fonction barrière.

Une fonction noyau est dite qualifiée si elle vérifie le lemme suivant :

Lemme 2.1.2 [12] *Si ψ satisfait les propriétés suivantes :*

- (I) $t \psi''(t) + \psi'(t) > 0, \quad \forall t < 1.$
- (II) $t \psi''(t) - \psi'(t) > 0, \quad \forall t > 1.$
- (III) $\psi'''(t) < 0, \quad \forall t > 0,$
- (IV) $2(\psi''(t))^2 - \psi'(t)\psi'''(t) > 0, \quad \forall t < 1$
- (V) $\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \forall t > 1, \forall \beta > 1.$

Alors ψ est dite qualifiée.

Présentation de la méthode

Dans ce paragraphe, nous considérons le problème (P) et son dual (D), d'un point de vue théorique et sans perte de généralité, on peut supposer que (CPI) est satisfaite. Nous réduisons μ et on résout le système de Newton suivant :

$$\begin{cases} A \Delta x & = 0 \\ A^T \Delta y + \Delta z & = 0 \\ z \Delta x + x \Delta z & = \mu e - xz. \end{cases} \quad (1)$$

Le nouvel itéré de la méthode de Newton est donné par :

$$x^+ = x + \alpha\Delta x, y^+ = y + \alpha\Delta y \text{ et } z^+ = z + \alpha\Delta z.$$

Maintenant pour faciliter l'étude de ces méthodes, on définit les vecteurs :

$$v = \sqrt{\frac{xz}{\mu}} \text{ et } v^{-1} = \sqrt{\left(\frac{xz}{\mu}\right)^{-1}}. \quad (2)$$

On pose aussi :

$$dx = \frac{v\Delta x}{x}, \quad dz = \frac{v\Delta z}{z}. \quad (3)$$

Avec cette notation, le système (1) devient :

$$\begin{cases} \bar{A} dx & = 0 \\ \bar{A}^T \Delta y + dz & = 0 \\ dx + dz & = v^{-1} - v, \end{cases} \quad (4)$$

tels que : $\bar{A} = \frac{1}{\mu}AV^{-1}X$, $V = \text{diag}(v)$ et $X = \text{diag}(x)$.

Notons ici que la partie droite de la troisième équation du système (4) n'est autre que l'opposé du gradient de la fonction barrière logarithmique Ψ , c.à.d,

$$dx + dz = -\nabla\Psi(v),$$

alors le système (4) peut être réécrit comme suit :

$$\begin{cases} \bar{A} dx & = 0 \\ \bar{A}^T \Delta y + dz & = 0 \\ dx + dz & = -\nabla\Psi(v). \end{cases}, \quad (5)$$

où la fonction barrière $\Psi : \mathbb{R}_{++}^n \rightarrow \mathbb{R}_+$ est définie comme suit :

$$\Psi(v) = \sum_{i=1}^n \psi(v_i). \quad (6)$$

avec

$$\psi(v_i) = \frac{v_i^2 - 1}{2} - \log v_i. \quad (7)$$

On peut vérifier que la fonction ψ définie par (7), est une fonction noyau, c'est la fonction noyau associée à la fonction barrière logarithmique Ψ . Notons que la paire (x, z) coïncide avec le μ -centre $(x(\mu), z(\mu))$ si et seulement si $v = e$.

Mesure de proximité : La fonction Ψ est dite fonction de proximité. Elle sert à mesurer la distance entre l'itéré et la trajectoire centrale (μ -centre) pour $\mu > 0$ donné. Nous définissons également la mesure de proximité, basée sur la norme, dite norme-proximité, $\delta : \mathbb{R}_{++}^n \rightarrow \mathbb{R}_+$ comme suit :

$$\delta(v) = \frac{1}{2} \|\nabla \Psi(v)\| = \frac{1}{2} \|dx + dz\| \quad (8)$$

Début algorithme 3

Données : Une fonction noyau ψ , un paramètre seuil $\tau > 1$, un paramètre de précision $\epsilon > 0$, un paramètre de mise à jour θ , $0 < \theta < 1$;

Initialisation : Soit (x^0, y^0, z^0) vérifie la (CPI), $\mu^0 > 0$, $v^0 = \sqrt{\frac{x^0 z^0}{\mu^0}}$;

Tant que : $n\mu > \epsilon$, faire

Début (itération externe) :

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

Tant que : $\Psi(v) > \tau$, faire

Début (itération interne) :

Résoudre le système (5) pour déterminer $(dx, \Delta y, dz)$ puis $(\Delta x, \Delta y, \Delta z)$;

Calculer le pas de déplacement α ;

$$\text{Poser : } x = x + \alpha \Delta x, y = y + \alpha \Delta y, z = z + \alpha \Delta z, v = \sqrt{\frac{xz}{\mu}}$$

Fin (itération interne) ;

Fin (itération externe) ;

Fin.

Algorithme 3 : Méthodes (TC) via une fonction noyau pour (PL)

Analyse de la complexité algorithmique :

Lemme 2.1.3 ([46]) : Si $\alpha \in [0, 1]$ alors $(1 + t)^\alpha \leq 1 + \alpha t, \forall t \geq -1$.

Lemme 2.1.4 ([46]) : Soit t_0, t_1, \dots, t_k , une suite des nombres positifs qui vérifie

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

tels que $\beta > 0$ et $0 < \gamma \leq 1$, alors : $K \leq \left\lceil \frac{t_0^\gamma}{\beta\gamma} \right\rceil$.

D'après [46], K désigne le nombre des itérations internes de chaque itération externe, il est défini comme suit :

$$K \leq \frac{[\Psi_0 - \tau]^\gamma}{\beta\gamma} \leq \frac{\Psi_0^\gamma}{\beta\gamma}, \quad (9)$$

avec : Ψ_0 est la première μ -mise à jour de Ψ et Ψ_k , $k = 1, 2, \dots, K - 1$, est une suite des valeurs de Ψ dans les itérations internes. La détermination du nombre total des itérations nécessaires pour trouver une solution optimale primale-duale nécessite le calcul du nombre des itérations externes.

Lemme 2.1.5 (*Lemme II.17, page 116, [51]*) *Soit k le nombre des itérations externes nécessaires pour trouver une solution optimale primale-duale approchée à une précision $\epsilon > 0$, alors on a :*

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

Le nombre total des itérations nécessaires pour trouver une solution optimale primale-duale approchée avec des précisions $\epsilon > 0$ et $\tau > 1$ n'est autre que la multiplication du nombre des itérations externes par le nombre des itérations internes. Nous obtenons le nombre total des itérations par l'expression suivante :

$$\frac{\Psi_0^\gamma}{\beta\gamma\theta} \log \frac{n}{\epsilon}. \quad (10)$$

Méthodes à petit pas : Les algorithmes qui utilisent une réduction θ en fonction de n s'appellent les algorithmes à petit pas. Si on prend $\theta = \mathcal{O}(\frac{1}{\sqrt{n}})$ comme facteur de réduction de μ et $\tau = \mathcal{O}(1)$, on s'assure d'une part de rester dans un voisinage de la trajectoire centrale et d'autre part d'une complexité algorithmique d'ordre $\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$, c'est la meilleure complexité obtenue à nos jours.

Méthodes à grand pas : Les algorithmes qui utilisent une réduction θ indépendante de n s'appellent les algorithmes à grand pas.

2.2 Méthodes de trajectoire centrale pour la programmation semi-définie

Dans cette section, nous aborderons l'étude des problèmes de minimisation de fonctions linéaires à contraintes linéaires, où la variable, est une matrice carrée symétrique $X \in S_n^+$. Il s'agit de résoudre le problème standard suivant :

$$(SDP) \begin{cases} \min C \bullet X \\ A_i \bullet X = b_i, \quad i = 1, \dots, m, \\ X \in S_n^+, \end{cases}$$

où :

C et $A_i, i = 1, \dots, m$, sont des matrices dans $S_n, b \in \mathbb{R}^m$.

Le programme semi-défini (SDP) généralise les problèmes de programmation linéaire (les vecteurs sont remplacés par des matrices), en particulier si la matrice X est diagonale, on trouve le programme linéaire (PL) classique. De plus, plusieurs problèmes de programmation non linéaire et quadratiques peuvent se reformuler sous forme (SDP). La contrainte de semi-définie positivité dans le programme semi-défini (SDP) induit un nombre infini d'équations contrairement au cas des programmes linéaires (PL) où la contrainte de positivité s'exprime par un nombre fini d'équations. Pour cela, les méthodes de points intérieurs sont très favorables vis à vis de la méthode du simplexe. Cette dernière est indésirable pour résoudre le programme semi-défini (SDP) puisque l'ensemble des contraintes est un cône convexe non polyédrique. Pour illustrer l'application des problèmes semi-définis dans différents problèmes, on donne quelques exemples. Le lecteur peut consulter [15] pour plus de détails.

Exemple 2 (Problème de programmation non linéaire) *On donne le programme*

non linéaire suivant :

$$\begin{cases} \min(x_1^3 + x_2) \\ x_1^3 x_2 \geq 1, \\ x_1 \geq 0, \quad x_2 \geq 0. \end{cases}$$

Ce problème s'écrit sous la forme d'un problème de (SDP) comme suit :

$$\begin{cases} \min C \bullet X \\ A_1 \bullet X = b_1, \\ X \in S_n^+, \end{cases}$$

avec

$$C = I, \quad X = \begin{pmatrix} x_1^3 & x_3 \\ x_3 & x_2 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}, \quad b_1 = 1.$$

Exemple 3 (Problème de coupure maximale) Dans la théorie des graphes, le problème de coupure s'écrit sous la forme suivante :

$$(CM) \begin{cases} \max \frac{1}{4} x^T L x \\ x \in \{-1, 1\}^n, \end{cases}$$

où $x \in \mathbb{R}^n$, est un vecteur qui représente une coupure S du graphe. Soit

$$L = \text{diag}(Ae) - A,$$

est une matrice $(n \times n)$ laplacienne associée au graphe et A une $(n \times n)$ matrice adjacente du graphe. Posons : $X = \frac{1}{4} x x^T$, alors

$$\frac{1}{4} x^T L x = \frac{1}{4} \text{tr}(x^T L x) = \frac{1}{4} \text{tr}(x x^T L) = X \bullet L = X^T \bullet L = \text{tr}(LX).$$

De même $X = \frac{1}{4}xx^T$ et comme

$$x_i = \pm 1 \Rightarrow xx^T = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & 1 & \vdots \\ 1 & \dots & 1 \end{pmatrix} \Rightarrow \text{diag}X = \frac{1}{4}e.$$

Le problème (CM) s'écrit sous forme d'un problème semi-défini (SDP) comme suit :

$$(SDP) = \begin{cases} \max X \bullet L \\ A_i \bullet X = \frac{1}{4}, & i = 1, \dots, n, \\ X \in S_n^+ , \end{cases}$$

où $X = \frac{1}{4}xx^T$ et $A_i = e_i e_i^T$ telle que e_i est la base canonique de \mathbb{R}^n . Le problème (SDP) obtenu est un problème continu. Donc on passe d'un cas discret (CM) à un cas continu (SDP) ce qui représente un avantage considérable en théorie et même en pratique.

Définition 2.2.1 Une matrice $X \in S_n$ est dite réalisable (resp. strictement réalisable) pour (SDP) si :

$$A_i \bullet X = b_i, \quad i = 1, \dots, m \quad \text{et} \quad X \in S_n^+$$

(resp. $A_i \bullet X = b_i, \quad i = 1, \dots, m$ et $X \in S_n^{++}$). On note $\mathcal{F}_{(SDP)}$ (resp. $\mathcal{F}_{(SDP)}^0$) l'ensemble des solutions réalisables (resp. l'ensemble des solutions strictement réalisables) pour (SDP).

Définition 2.2.2 La valeur optimale primale de (SDP) est définie par :

$$m_p = \min \{ C \bullet X, \quad A_i \bullet X = b_i, i = 1, \dots, m, \quad X \in S_n^+ \}.$$

X^* est une solution optimale primale de (SDP) si :

$$X^* \in \mathcal{F}_{(SDP)} \quad \text{et} \quad C \bullet X^* = m_p.$$

2.2.1 Dualité en programmation semi-définie

La dualité en (*SDP*) est très similaire à la dualité classique en programmation linéaire à quelques différences près [55, 15]. Pour obtenir le problème dual de (*SDP*), on considère la fonction duale,

$$\begin{aligned} q(y) &= \min_{X \in S_n^+} \left[C \bullet X + \sum_{i=1}^m (b_i - A_i \bullet X) y_i, y \in \mathbb{R}^m \right] \\ &= \min_{X \in S_n^+} \left[(C - \sum_{i=1}^m y_i A_i) \bullet X \right] + \sum_{i=1}^m y_i b_i. \end{aligned}$$

D'où

$$\max_{y \in \mathbb{R}^m} q(y) = \begin{cases} \max b^T y & \text{si } \left(C - \sum_{i=1}^m y_i A_i \right) \in S_n^+ \\ -\infty & \text{ailleurs .} \end{cases}$$

Le problème dual de (*SDP*) est donc :

$$\begin{cases} \max b^T y \\ C - \sum_{i=1}^m y_i A_i \in S_n^+ \\ y \in \mathbb{R}^m \end{cases} \Leftrightarrow \begin{cases} \max b^T y \\ C - \sum_{i=1}^m y_i A_i = Z \\ y \in \mathbb{R}^m, Z \in S_n^+ \end{cases} . \quad (DSDP)$$

Définition 2.2.3 Une solution réalisable de (*DSDP*) est le couple $(y, Z) \in \mathbb{R}^m \times S_n^+$ tel que :

$$C - \sum_{i=1}^m y_i A_i = Z.$$

De même le couple (y, Z) est dit strictement réalisable pour (*DSDP*) si :

$$(y, Z) \in \mathbb{R}^m \times S_n^{++}.$$

On note $\mathcal{F}_{(DSDP)}$ (resp. $\mathcal{F}_{(DSDP)}^0$) l'ensemble des solutions réalisables pour (*DSDP*) (resp. l'ensemble des solutions strictement réalisables pour (*DSDP*)).

Définition 2.2.4 La valeur optimale de (DSDP) est définie par :

$$m_d = \max_{y \in \mathbb{R}^m} \left\{ b^T y : C - \sum_{i=1}^m y_i A_i \in S_n^+ \right\}$$

avec

$$(y^*, Z^*) \in \mathcal{F}_{(DSDP)}, \quad b^T y^* = m_d$$

et

$$Z^* = C - \sum_{i=1}^m y_i^* A_i.$$

On a le résultat de dualité faible suivant :

Théorème 2.2.1 (Dualité faible) [55] Si X et (y, Z) sont des solutions de (SDP) et (DSDP) respectivement, alors on a toujours

$$C \bullet X - b^T y = Z \bullet X \geq 0,$$

cette différence est appelée saut de dualité.

Preuve 1 On a

$$\begin{aligned} C \bullet X - b^T y &= C \bullet X - \sum_{i=1}^m y_i b_i \\ &= C \bullet X - \sum_{i=1}^m y_i (A_i \bullet X) \\ &= (C - \sum_{i=1}^m y_i A_i) \bullet X \\ &= Z \bullet X = \text{tr}(ZX) \geq 0. \end{aligned}$$

□

Remarque 2.2.1 Contrairement à la programmation linéaire, il n'est pas toujours vrai que l'optimalité de (SDP) et (DSDP) implique que $Z \bullet X = 0$, comme le montre l'exemple suivant :

Exemple 4 [15] $E = S_3$ l'espace des matrices symétriques 3×3 :

$$m_p = \min_X [C \bullet X : X \in S_3^+, A_i \bullet X = b_i; \quad i = 1, 2] \quad (SDP)$$

où

$$C = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

$$X = \begin{pmatrix} x_{11} & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix}, \quad b = (0, -1)^T.$$

Ce problème s'écrit aussi sous la forme suivante :

$$m_p = \min_X \left[x_{12} : \begin{pmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{pmatrix} \in S_3^+ \right]. \quad (SDP)$$

L'ensemble des solutions réalisables implique : $x_{12} = 0$ et $x_{22} \geq 0$.

En outre l'ensemble des solutions optimales est :

$$\mathcal{F}_{opt}(SDP) = \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & x_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} ; x_{22} \geq 0 \right\}.$$

Il s'ensuit que $m_p = 0$.

Ecrivons le problème dual

$$m_d = \max_{y \in \mathbb{R}^2} \left\{ b^T y : C - \sum_{i=1}^m y_i A_i \in S_3^+ \right\} \quad (DSDP)$$

qui correspond à la forme :

$$m_d = \max_{y \in \mathbb{R}^2} \left\{ -y_2 : \begin{pmatrix} -y_1 & \frac{1-y_2}{2} & 0 \\ \frac{1-y_2}{2} & 0 & 0 \\ 0 & 0 & y_2 \end{pmatrix} \in S_3^+ \right\}. \quad (DSDP)$$

L'ensemble des solutions réalisables correspond à l'ensemble

$$\{(y_1, y_2) \in \mathbb{R}^2 : y_1 \leq 0, y_2 = 1\}$$

En outre l'ensemble des solutions optimales est :

$$\mathcal{F}_{opt}(DSDP) = \{(y_1, 1) ; y_1 \leq 0\}.$$

Il s'ensuit que $m_d = -1 \neq m_p = 0$. Les problèmes (SDP) et (DSDP) sont tous les deux réalisables ($\mathcal{F}_{(SDP)}$ et $\mathcal{F}_{(DSDP)}$ sont tous les deux non vides) mais, $-\infty < m_d < m_p < +\infty$.

On revient maintenant aux conditions qui assurent la dualité forte, et l'existence des solutions primales et duales.

Théorème 2.2.2 [44]

1. Si le problème (SDP) est strictement réalisable, c-à-d :

$$\exists X \in S_n^{++} : A_i \bullet X = b_i, i = 1, \dots, m,$$

alors, $m_d = m_p$.

2. Si en outre m_p est fini, alors l'ensemble $\mathcal{F}_{opt}(DSDP)$ est compact non vide.
3. Si le problème (DSDP) est strictement réalisable, c-à-d :

$$\exists y \in \mathbb{R}^m : C - \sum_{i=1}^m y_i A_i \in S_n^{++}$$

alors, $m_d = m_p$.

4. Si en outre m_d est fini, alors l'ensemble $\mathcal{F}_{\text{opt}(SDP)}$ est compact non vide.

2.2.2 Complémentarité en SDP

Comme dans la programmation linéaire, on peut exprimer la condition d'optimalité de (SDP) et $(DSDP)$ sous la forme d'une condition dite de complémentarité.

Théorème 2.2.3 Soient $X^* \in \mathcal{F}_{(SDP)}^0$ et $(y^*, Z^*) \in \mathcal{F}_{(DSDP)}^0$ de saut de dualité

$$C \bullet X^* - b^T y^* = Z^* \bullet X^*$$

Alors X^* et (y^*, Z^*) sont des solutions optimales pour (SDP) et $(DSDP)$ respectivement si et seulement si $X^* Z^* = 0$.

Le problème de complémentarité s'écrit comme suit :

$$\begin{cases} A_i \bullet X = b_i, & i = 1, \dots, m, & X \succeq 0, \\ C - \sum_{i=1}^m y_i A_i = Z, & Z \succeq 0, \\ XZ = 0. \end{cases} \quad (PC)$$

Notons que le produit $x_i z_i$ dans la programmation linéaire est remplacé par la multiplication ordinaire des deux matrices XZ dans la programmation semi-définie. Ce problème fait l'objet d'études de recherche acharnées essentiellement sur les approches Newtoniennes [7], [42].

Remarque 2.2.2 Comme (SDP) est un problème convexe, la solution du problème de complémentarité est un optimum global pour (SDP) .

2.2.3 Conditions de Karush - Kuhn - Tucker (KKT)

Comme pour la programmation linéaire, les conditions d'optimalité (KKT) du problème primal et de son dual sont identiques. Elles sont de plus nécessaires et suffisantes, à cause de la convexité du problème, et s'écrivent

$$(KKT) \begin{cases} A_i \bullet X = b_i & \text{pour } i = 1, \dots, m \\ \sum_{i=1}^m y_i A_i + Z = C, \\ XZ = 0, \\ X \succeq 0 \text{ et } Z \succeq 0. \end{cases}$$

La seule équation non linéaire dans le système de (KKT) est $XZ = 0$. La troisième condition est équivalente à $X \bullet Z = 0$. Les méthodes de points intérieurs en programmation semi-définie vont donc tenter de résoudre ce système par la méthode de Newton.

2.2.4 Fonction barrière

En programmation mathématique, quand on traite un problème de minimisation sous contraintes égalité et inégalité définissant un domaine réalisable, il peut être intéressant de se ramener à un problème ne comportant que des contraintes égalité en substituant aux contraintes inégalité un terme additionnel (appelé terme barrière) à la fonction objectif pondéré par un paramètre réel positif μ , où $\mu > 0$ est le paramètre de barrière. Le terme barrière possède la propriété fondamentale de tendre vers $+\infty$ quand un point test réalisable X approche la frontière du domaine réalisable.

La théorie de Nesterov et Nemirovsky [44], nous apprend qu'une telle barrière doit être auto-concordante pour être efficace (c'est-à-dire entraîner une complexité polynômiale). La plus simple et la plus élégante de ces barrières est définie par

$$\Phi(X) = -\ln \det X,$$

(où $\det X$ est le déterminant de la matrice X). En effet, lorsqu'on s'approche de la frontière, une des valeurs propres de X doit tendre vers zéro, ce qui fait également tendre le déterminant vers zéro et Φ vers $+\infty$.

2.2.5 Trajectoire centrale et mesure de dualité

La solution du problème auquel on a adjoint la fonction barrière peut encore s'interpréter comme un point de la trajectoire centrale, à l'aide des conditions (*KKT*). Pour une valeur donnée du paramètre barrière μ , il est défini par

$$\left\{ \begin{array}{l} A_i \bullet X = b_i \quad \text{pour } i = 1, \dots, m \\ \sum_{i=1}^m y_i A_i + Z = C, \\ XZ = \mu I, \\ X \succ 0 \text{ et } Z \succ 0. \end{array} \right.$$

On a donc remplacé la condition de complémentarité $XZ = 0$ par la relaxation $XZ = \mu I$. Afin de mesurer la proximité par rapport à la trajectoire centrale, on définit aussi une mesure de dualité μ par :

$$\mu(X, y, Z) = \frac{1}{n} X \bullet Z,$$

Les voisinages de la trajectoire centrale sont définis d'une manière analogue à leurs équivalents du cas de la programmation linéaire.

Mise en parallèle des concepts avec le cas linéaire

Il est possible d'établir un parallèle entre les concepts de la programmation semi-définie et ceux de la programmation linéaire à l'aide de la remarque suivante :

Remarque 2.2.3 *Les équations de la programmation semi-définie sont équivalentes à celles de la programmation linéaire où l'on aurait remplacé les matrices $M \in S_n$ par λ_M (le vecteur formé des valeurs propres de la matrice M , rangées en ordre décroissant).*

Ainsi, la contrainte $X \succeq 0$ (ou $X \succ 0$) est bien équivalente à $\lambda_X \geq 0$ (ou $\lambda_X > 0$). De même, la condition $XZ = \mu I$ est équivalente à $\lambda_{XZ} = \mu e$ ou $(\lambda_{XZ})_i = \mu$ pour $i = 1, \dots, n$, et l'on retrouve la contrainte $x_i z_i = \mu$.

La mesure de dualité μ est définie par :

$$\mu(X, y, Z) = \frac{1}{n} X \bullet Z = \frac{1}{n} \text{tr}(XZ) = \frac{1}{n} \sum_{i=1}^n (\lambda_{XZ})_i,$$

Enfin, la fonction barrière

$$\Phi(X) = -\log \det X = -\log \prod_{i=1}^n \lambda_{X_i} = -\sum_{i=1}^n \log \lambda_{X_i},$$

est à mettre en parallèle avec

$$\Phi(X) = -\log \prod_{i=1}^n \lambda_{X_i} = -\sum_{i=1}^n \log x_i.$$

2.2.6 Description de la méthode de TC

On peut établir une classification des méthodes de points intérieurs en programmation semi-définie identique à celle déjà vue pour le cas linéaire (méthodes primales, duales, primales-duales, à départ admissible ou pas, etc.). Ici aussi, on constate que les méthodes primales-duales sont les plus efficaces théoriquement et numériquement. Les méthodes affines, de trajectoire centrale et de réduction de potentiel se transposent intégralement à la programmation semi-définie. Ainsi, les algorithmes décrits dans la section 2.1 peuvent être suivis à la lettre, à condition de remplacer x par X et z par Z . Le calcul de μ s'effectue selon l'équation définie plus haut, et le nouveau système (SL_μ) à résoudre se

présente sous la forme suivante :

$$\begin{cases} A_i \bullet \Delta X = 0, & i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z = 0, \\ Z\Delta X + X\Delta Z = \mu I - XZ. \end{cases} \quad (11)$$

Il faut remarquer deux différences importantes par rapport au système (11) du cas linéaire. D'une part, bien que linéaire, ce système n'est pas sous la forme $Ax = b$, et ne peut donc plus se résoudre directement par les méthodes classiques applicables aux systèmes linéaires (Cholesky, gradient conjugué, etc...). Les logiciels devront donc le transformer avant de lui appliquer une de ces méthodes. D'autre part, il existe une autre difficulté plus importante, parce qu'elle se situe au niveau conceptuel. Les matrices X et Z appartenant à S_n , pour que $X + \Delta X$ et $Z + \Delta Z$ appartiennent également à S_n , il faut que ΔX et ΔZ soient symétriques. Or, comme le produit XZ n'est pas symétrique, on constate que la composante ΔX fournie par le système ne le sera pas non plus (ΔZ l'est à cause de la seconde équation). La façon la plus simple de surmonter ce problème est de rendre symétrique la composante ΔX après chaque itération (par exemple en prenant $\frac{(\Delta X + \Delta X^T)}{2}$). Cette façon de procéder est connue sous le nom de méthode XZ .

Certains auteurs préfèrent cependant modifier directement le système d'équations pour qu'il fournisse un ΔX symétrique. A cet effet, on définit un opérateur de symétrisation H_P pour toute matrice non singulière P selon [58]

$$H_P(M) = \frac{(PMP^{-1} + (PMP^{-1})^T)}{2}.$$

On reformule alors le système de Newton selon

$$\begin{cases} A_i \bullet \Delta X & = 0, & i = 1, \dots, m \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z & = 0, \\ H_P(Z\Delta X + X\Delta Z) & = \mu I - H_P(XZ), \end{cases}$$

ce qui nous donne un pas $(\Delta X, \Delta y, \Delta Z)$ totalement symétrique. Parmi les choix possibles pour P , citons $P = I$ (méthode *AHO* [8]), $P = Z^{\frac{1}{2}}$ [58] ou encore $P^T P = X^{-1}$ ou Z (Monteiro [42]). Aucune de ces méthodes ne s'est encore affirmée comme clairement préférable aux autres, et de nombreux chercheurs étudient d'autres possibilités. Tandis que Todd et al. [53], ont montré que pour symétriser le système (11), on introduit directement une matrice P inversible comme suit :

$$\begin{cases} A_i \bullet \Delta X & = 0, & i = 1, 2, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z & = 0, \\ \Delta X + P \Delta Z P^T & = \mu Z^{-1} - X, & X, Z \succeq 0. \end{cases} \quad (12)$$

Les directions de Nesterov Todd :

L'un des choix de la matrice P est :

$$P = X^{\frac{1}{2}} (X^{\frac{1}{2}} Z X^{\frac{1}{2}})^{-\frac{1}{2}} X^{\frac{1}{2}} = Z^{-\frac{1}{2}} (Z^{\frac{1}{2}} X Z^{\frac{1}{2}})^{\frac{1}{2}} Z^{-\frac{1}{2}} \quad (\text{Nesterov Todd scaling matrix})$$

où $P \in S_n^{++}$.

Maintenant, on définit la matrice symétrique définie positive $D = P^{\frac{1}{2}}$. Le rôle de D est de mettre à l'échelle les deux matrices X et Z en une même matrice symétrique, définie positive V comme suit :

$$V = \frac{1}{\sqrt{\mu}} D^{-1} X D^{-1} = \frac{1}{\sqrt{\mu}} D Z D = \frac{1}{\sqrt{\mu}} (D^{-1} X Z D)^{\frac{1}{2}}. \quad (13)$$

De plus les nouvelles directions (scaling direction) D_X et D_Z sont définies par :

$$D_X = \frac{1}{\sqrt{\mu}} D^{-1} \Delta X D^{-1}, D_Z = \frac{1}{\sqrt{\mu}} D \Delta Z D. \quad (14)$$

Alors le système (12) devient :

$$\begin{cases} \bar{A}_i \bullet D_{\mathbf{X}} = 0, & i = 1, 2, \dots, m \\ \sum_{i=1}^n \Delta y_i \bar{A}_i + D_{\mathbf{Z}} = 0, \\ D_X + D_Z = V^{-1} - V, \end{cases} \quad (15)$$

où $\bar{A}_i = \frac{1}{\sqrt{\mu}} D A_i D$, $i = 1, 2, \dots, m$.

On note que les matrices $[\bar{A}_1, \bar{A}_2, \dots, \bar{A}_m]$ sont aussi linéairement indépendantes, les directions D_X, D_Z sont symétriques et orthogonales ($D_X \bullet D_Z = 0$), qui provient des première et seconde équations de (15) ou de l'orthogonalité de ΔX et ΔZ .

On définit la fonction barrière Ψ comme suit :

Définition 2.2.5 [46]

$$\Psi(V) = \sum_{i=1}^n \psi(\lambda_i(V)) = Tr(\psi(V)). \quad (16)$$

Suivant [45, 46] nous décrivons l'approche basée sur les fonctions noyaux à (*SDP*). Étant donné la fonction noyau ψ et on associe $\psi(V)$ et $\psi'(V)$ comme dans le théorème 1.2.1, on remplace le membre droit de la troisième équation dans (15) par $-\psi'(V)$. Nous considérons donc le système suivant.

$$\begin{cases} \bar{A}_i \bullet D_X = 0, & i = 1, 2, \dots, m, \\ \sum_{i=1}^n \Delta y_i \bar{A}_i + D_Z = 0, \\ D_X + D_Z = -\nabla \Psi(\mathbf{V}). \end{cases} \quad (17)$$

Ayant $D_X, \Delta y$ et D_Z comme solutions, ΔX et ΔZ peuvent être calculées à partir de (14). Du fait que $D_X \perp D_Z$, alors

$$D_X \bullet D_Z = D_Z \bullet D_X = 0. \quad (18)$$

On a :

$$D_X = D_Z = 0 \Leftrightarrow \nabla \Psi(V) = 0 \Leftrightarrow V = I \Leftrightarrow X = X(\mu), Z = Z(\mu). \quad (19)$$

Nous utilisons Ψ comme fonction de proximité pour mesurer la distance entre l'itéré en cours et le point centre pour $\mu > 0$. Nous définissons aussi la mesure de proximité basée sur la norme, $\delta : S_n^{++} \times S_n^{++} \times \mathbb{R}_{++} \rightarrow \mathbb{R}_+$, comme suit :

$$\delta(X, Z, \mu) = \delta(V) = \frac{1}{2} \|\nabla \Psi(V)\| = \frac{1}{2} \|D_X + D_Z\|. \quad (20)$$

Début algorithme 4

Initialisation : Une fonction noyau ψ , un paramètre seuil $\tau > 1$;

un paramètre de précision $\epsilon > 0$, un paramètre de mise à jour θ ;

$0 < \theta < 1$, (X^0, y^0, Z^0) vérifie la (CPI), $\mu^0 > 0$;

Tant que : $n\mu > \epsilon$, faire

Début (itération externe) :

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

Tant que : $\Psi(V) > \tau$, faire

Début (itération interne) :

Résoudre le système (15) pour déterminer $(D_X, \Delta y, D_Z)$ puis $(\Delta X, \Delta y, \Delta Z)$;

Calculer le pas de déplacement α ;

$X = X + \alpha \Delta X$, $y = y + \alpha \Delta y$;

$Z = Z + \alpha \Delta Z$, $V = \frac{1}{\sqrt{\mu}} (D^{-1} X Z D)^{\frac{1}{2}}$;

Fin tant que (itération interne) ;

Fin tant que (itération externe) ;

Fin.

Algorithme 4 : Méthodes (TC) via une fonction noyau pour (SDP)

Comme dans le cas linéaire les paramètres τ , θ et le pas de déplacement α doivent être choisis de telle sorte que l'algorithme soit minimale «optimisé». Dans ce sens, le nombre

d'itérations requis par l'algorithme est aussi petit que possible.

Chapitre 3

Méthodes de (TC) pour (PL) et (SDP) basées sur une nouvelle fonction noyau

Dans ce chapitre on propose une méthode de points intérieurs de trajectoire centrale (*TC*) de type primal-dual pour résoudre (*PL*) et (*SDP*). La dite méthode est basée sur l'introduction de nouvelles fonctions dites auto-régulières introduites récemment par Peng, Roos et Terlaky en [47] inclus les fonctions noyaux utilisées par Bai et Roos [11], Bai, Roos et El Ghami [12] pour résoudre un programme linéaire et semi-défini. Ces méthodes, néanmoins, déterminent une nouvelle classe de directions de Newton mais elles sont aussi considérées comme une proximité pour mesurer la qualité des points par rapport à la trajectoire centrale. Ces dernières fonctions satisfont quelques propriétés qui conduisent à un modèle très important pour développer ces méthodes. Ce nouveau paradigme donne naissance à une nouvelle classe de méthodes de points intérieurs de (*TC*) dites : "méthodes barrières généralisées". Contrairement aux méthodes barrières classiques de points intérieurs qui sont basées sur la fonction barrière logarithmique (voir [12]), celles-ci sont basées sur la notion des fonctions auto-régulières. Historiquement, presque tous les résultats de la complexité polynomiale des algorithmes primaux-duaux

de (TC) et leurs implémentations numériques sont analysés moyennant l'approche barrière logarithmique. La complexité polynômiale trouvée par cette dernière est d'ordre $\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$ pour les algorithmes à petit pas et $\mathcal{O}(n \log \frac{n}{\epsilon})$ pour les algorithmes à grand pas [1]. Avec ces nouvelles méthodes la complexité polynômiale est améliorée de $\mathcal{O}(n \log \frac{n}{\epsilon})$ à $\mathcal{O}(\sqrt{n} \log n \log \frac{n}{\epsilon})$ pour les algorithmes à grand pas. Pour plus de détails sur ces méthodes voir l'article de Bai et al. [10, 13], et la thèse d'ELGhami [29]. Pour les algorithmes à petit pas, on conserve la même complexité. Sur le plan numérique ces algorithmes montrent leur efficacité par rapport aux algorithmes classiques.

Dans ce chapitre, on propose une nouvelle fonctions noyaux avec un terme barrière double définie par :

$$\psi(t) = \frac{t^2 - 1 - \log t}{2} + \frac{e^{\frac{1}{t^q} - 1} - 1}{2q} \text{ for } t > 0, q \geq 1. \quad (21)$$

Dans la première partie on l'utilise pour résoudre un programme linéaire, puis, on généralise l'étude dans la deuxième partie du cas SDP .

Première partie

Programmation linéaire

3.1 Propriétés de la nouvelle fonction noyau

Dans cette partie, nous donnons les propriétés de la nouvelle fonctions noyaux, qui sont essentielles à notre analyse de la complexité.

On vérifie facilement que :

- $\lim_{t \rightarrow 0} \psi(t) = \infty$.
- $\lim_{t \rightarrow +\infty} \psi(t) = \infty$.

Nous donnons les trois premières dérivées par rapport à t :

$$\begin{aligned} \psi'(t) &= t - \frac{1}{2t} - \frac{e^{\frac{1}{iq}-1}}{2t^{q+1}}, \\ \psi''(t) &= 1 + \frac{1}{2t^2} + \frac{1}{2} \left(\frac{(q+1)t^q + q}{t^{2q+2}} \right) e^{\frac{1}{iq}-1}, \\ \psi'''(t) &= \frac{-1}{t^3} - \frac{1}{2} (q^2 t^{-(3q+3)} + 3q(q+1)t^{-(2q+3)} + (q+1)(q+2)t^{-(q+3)}) e^{\frac{1}{iq}-1}. \end{aligned} \tag{22}$$

On a $\psi(1) = \psi'(1) = 0$ et $\psi''(t) > 1$, donc ψ est une fonction noyau.

Les trois prochains lemmes seront vérifiés si les trois conditions (I), (II) et (III) du Lemme 2.1.2 précédent sont satisfaites.

Pour montrer (I) et (II), on utilise (22) on obtient :

$$t\psi''(t) + \psi'(t) = 2t + \frac{1}{2} q(t^{-(q+1)} + t^{-(2q+1)}) e^{\frac{1}{iq}-1} > 0, \quad \forall t > 0,$$

$$t\psi'''(t) - \psi'(t) = \frac{1}{t} + \frac{1}{2t^{(2q+1)}} [(q+2)t^q + q] e^{\frac{1}{iq}-1} > 0, \quad \forall t > 0.$$

Il est clair de (22), que :

$$\psi'''(t) = -\left[\frac{1}{t^3} + \frac{1}{2} (q^2 t^{-(3q+3)} + 3q(q+1)t^{-(2q+3)} + (q+1)(q+2)t^{-(q+3)}) e^{\frac{1}{iq}-1} \right] < 0,$$

pour $t > 0$ et $q > 0$.

Lemme 3.1.1 [12] *Soit ψ une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes :*

- (a) $\psi\sqrt{t_1 t_2} \leq \frac{\psi(t_1) + \psi(t_2)}{2}$, pour tout $t_1, t_2 > 0$.
 (b) La fonction ϕ définie par $\phi(\xi) = \psi(e^\xi)$ est convexe.
 (c) $t\psi''(t) + \psi'(t) \geq 0, \forall t > 0$.

Preuve 2 Signalons que la fonction ϕ est continue donc pour qu'elle soit convexe, il suffit d'être mid-convexe.

Montrons que (a) \Leftrightarrow (b)

$$\begin{aligned} \psi\sqrt{t_1 t_2} &\leq \frac{\psi(t_1) + \psi(t_2)}{2}, \forall t_1, t_2 > 0. \text{ (on pose } t_i = e^{\xi_i}, i = 1, 2) \\ \Leftrightarrow \psi\left(\sqrt{e^{\xi_1} e^{\xi_2}}\right) &= \phi\left(\frac{\xi_1 + \xi_2}{2}\right) \leq \frac{\psi(e^{\xi_1}) + \psi(e^{\xi_2})}{2} = \frac{\phi(\xi_1) + \phi(\xi_2)}{2} \\ \Leftrightarrow \phi &\text{ est mid-convexe.} \end{aligned}$$

Concernant l'équivalence (b) \Leftrightarrow (c)

$$\begin{aligned} \phi \text{ est convexe} &\Leftrightarrow \phi''(\xi) = [\psi(e^\xi)]'' = [\psi'(e^\xi) + e^\xi \psi''(e^\xi)] e^\xi \geq 0, \forall \xi \in \mathbb{R} \\ &\Leftrightarrow [\psi'(t) + t\psi''(t)] t \geq 0, \forall t > 0 \text{ tel que } t = e^\xi \\ &\Leftrightarrow [\psi'(t) + t\psi''(t)] \geq 0, \forall t > 0. \end{aligned}$$

Ce qui termine la démonstration. \square

Lemme 3.1.2 [12] Soit ψ une fonction deux fois différentiable pour $t > 0$, alors les trois propriétés suivantes sont équivalentes :

- (a) $\psi\sqrt{\frac{t_1^2 + t_2^2}{2}} \leq \frac{\psi(t_1) + \psi(t_2)}{2}$, pour tout $t_1, t_2 > 0$.
 (b) $\psi(\sqrt{\xi})$ est convexe, $\xi > 0$.
 (c) $t\psi''(t) - \psi'(t) \geq 0, t > 0$.

Preuve 3 (b) \Leftrightarrow (a) :

La fonction $\psi(\sqrt{\xi})$ est convexe si et seulement si $\forall \xi_1, \xi_2 \in \mathbb{R}^+$, on a

$$\psi\left(\sqrt{\frac{\xi_1 + \xi_2}{2}}\right) \leq \frac{\psi(\sqrt{\xi_1}) + \psi(\sqrt{\xi_2})}{2}.$$

On pose $t_1 = \sqrt{\xi_1}$ et $t_2 = \sqrt{\xi_2}$, l'inégalité précédente devient

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

(b) \Leftrightarrow (c) :

$$\psi \left(\sqrt{\xi} \right) \text{ est convexe} \Leftrightarrow \left(\psi \left(\sqrt{\xi} \right) \right)'' = \frac{1}{4\xi^{\frac{3}{2}}} \left(\sqrt{\xi} \psi'' \left(\sqrt{\xi} \right) - \psi' \left(\sqrt{\xi} \right) \right) \geq 0.$$

On pose $t = \sqrt{\xi}$, alors

$$\frac{-1}{4t^3} (t\psi''(t) - \psi'(t)) \geq 0 \Leftrightarrow (t\psi''(t) - \psi'(t)) \geq 0, \text{ pour } t > 0.$$

Ce qui termine la preuve. \square

Lemme 3.1.3 Si ψ vérifie la propriété (II) et (III) du Lemme 2.1.2, alors ψ vérifie :

$$\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \forall t > 1, \forall \beta > 1. \text{ (la propriété (V))}$$

Preuve 4 Supposons que ψ vérifie (II) et (III) du Lemme 2.1.2. Soit $t > 1$, on considère la fonction

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

on a

$$g'(\beta) = \psi'''(\beta t) [t\psi''(t) - \psi'(t)] - \beta t \psi'(t) \psi'''(\beta t) > 0, \forall \beta > 1,$$

donc la fonction g est strictement croissante et $g(1) = 0$, alors $g(\beta) > 0, \forall \beta > 1$.

Donc ψ est une fonction qualifiée. \square

Lemme 3.1.4 Pour ψ , on a :

$$\frac{1}{2}(t-1)^2 \leq \psi(t) \leq \frac{1}{2}(\psi'(t))^2, \quad t > 0, \tag{23}$$

$$\psi(t) \leq \frac{2+q}{2}(t-1)^2, \quad t > 1. \tag{24}$$

Preuve 5 Pour (23), du fait que ψ est une fonction noyau et $\psi'' > 1$, on a

$$\begin{aligned}\psi(t) &= \int_1^t \int_1^\xi \psi''(\zeta) d\zeta d\xi \\ &\leq \int_1^t \int_1^\xi \psi''(\xi) \psi''(\zeta) d\zeta d\xi \\ &= \int_1^t \psi''(\xi) \psi'(\xi) d\xi \\ &= \frac{1}{2} (\psi'(t))^2.\end{aligned}$$

Pour la deuxième inégalité on obtient :

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\zeta) d\zeta d\xi \geq \int_1^t \int_1^\xi 1 d\zeta d\xi = \frac{1}{2} (t-1)^2.$$

Pour (24), on utilise le Théorème de Taylor avec

$$\psi(1) = \psi'(1) = 0, \quad \psi''(1) = 2 + q, \quad \text{et} \quad \psi'''(t) < 0,$$

on a :

$$\begin{aligned}\psi(t) &= \psi(1) + \psi'(1)(t-1) + \frac{1}{2} \psi''(1)(t-1)^2 + \frac{1}{6} \psi'''(\xi)(\xi-1)^3 \\ &= \frac{1}{2} \psi''(1)(t-1)^2 + \frac{1}{6} \psi'''(\xi)(\xi-1)^3 \quad \text{pour } \xi, 1 \leq \xi \leq t \\ &\leq \frac{1}{2} \psi''(1)(t-1)^2 \\ &= \frac{2+q}{2} (t-1)^2.\end{aligned}$$

Ce qui achève la démonstration. \square

Lemme 3.1.5 Soit $\varrho : [0, \infty) \rightarrow [1, \infty)$, l'inverse de la fonction ψ pour $t \geq 1$.

Alors on a :

$$1 + \sqrt{\frac{2s}{q+2}} \leq \varrho(s) \leq 1 + \sqrt{2s}. \quad (25)$$

Preuve 6 Soit $s = \psi(t), t \geq 1$, c.à.d, $\varrho(s) = t, s \geq 0$. Par la définition de ψ on a

$$s = \psi(t) \geq \frac{1}{2}(t-1)^2.$$

Ceci implique que :

$$t = \varrho(s) \leq 1 + \sqrt{2s}.$$

De (24), on a :

$$s = \psi(t) \leq \frac{2+q}{2}(t-1)^2, \quad t > 1,$$

alors

$$t = \varrho(s) \geq 1 + \sqrt{\frac{2s}{q+2}}. \square$$

Dans le Lemme suivant on désigne par $\psi_b(t)$ le terme barrière de $\psi(t)$, donné par :

$$\psi_b(t) = \frac{-\log t}{2} + \frac{e^{\frac{1}{t^q}-1} - 1}{2q} \text{ pour } t > 0, q \geq 1. \quad (26)$$

Lemme 3.1.6 Soit $\rho : [0, \infty) \rightarrow (0, 1]$, la fonction inverse de la restriction de $-\frac{\psi'(t)}{2}$ dans l'intervalle $(0, 1]$, $\underline{\rho} : [0, \infty) \rightarrow (0, 1]$, être la fonction inverse de la restriction de $-\psi'_b(t)$ dans l'intervalle $(0, 1]$ et $s_b = -\psi'_b(t)$. Alors, on a :

$$\rho(s) \geq \underline{\rho}(1 + 2s).$$

$$\underline{\rho}(s_b) \geq \frac{1}{(\log(2s_b) + 1)^{\frac{1}{q}}}. \quad (27)$$

Preuve 7 Soit $t = \rho(s)$. De la définition de ρ comme fonction inverse de $-\frac{\psi'(t)}{2}$ pour $t \in (0, 1]$, cela signifie que

$$-2s = \psi'(t) = t + \psi'_b(t), \quad 0 < t \leq 1.$$

Puisque $t \leq 1$, ceci implique

$$-\psi'_b(t) = t + 2s \leq 1 + 2s.$$

Comme $-\psi'_b$ est strictement décroissante, $\underline{\rho}(s_b)$ est aussi strictement décroissante.

On a :

$$\rho(s) = t = \underline{\rho}(s_b) = \underline{\rho}(-\psi'_b(t)) \geq \underline{\rho}(1 + 2s).$$

Pour (27), soit

$$s_b = \frac{1}{2\underline{\rho}(s_b)} + \frac{1}{2}\underline{\rho}(s_b)^{-(q+1)}e^{\underline{\rho}(s_b)^{-q}-1}, \quad 0 < \underline{\rho}(s_b) \leq 1,$$

signifie que

$$e^{\underline{\rho}(s_b)^{-q}-1} = 2\underline{\rho}(s_b)^{(q+1)}s_b - \underline{\rho}(s_b)^q \leq 2s_b.$$

Par conséquent

$$\underline{\rho}(s_b) \geq \frac{1}{(\log(2s_b) + 1)^{\frac{1}{q}}}.$$

Ce qui achève la démonstration. \square

Lemme 3.1.7 Pour tout vecteur positif v et $\beta \geq 1$, on a

$$\Psi(\beta v) \leq n\psi(\beta \varrho(\frac{\Psi(v)}{n})).$$

Preuve 8 Utilisant le Lemme 3.1.3 et le Théorème 3.2 dans [12], on trouve le résultat. \square

Lemme 3.1.8 Soit $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$, si $\Psi(v) \leq \tau$, alors on a ;

$$\Psi(v_+) \leq \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)}.$$

Preuve 9 De $\frac{1}{\sqrt{1-\theta}} \geq 1$ et $\varrho(\frac{\Psi(v)}{n}) \geq 1$, alors $\frac{\varrho(\frac{\Psi(v)}{n})}{\sqrt{1-\theta}} \geq 1$. Pour $t \geq 1$, on a :

$$\psi(t) \leq \frac{t^2 - 1}{2}.$$

Utilisant le Lemme 3.1.7 avec $\beta = \frac{1}{\sqrt{1-\theta}}$, (25) et $\Psi(v) \leq \tau$, on trouve

$$\begin{aligned} \Psi(v_+) &\leq n\psi\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right)\right) \\ &\leq \frac{n}{2} \left(\left[\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right) \right]^2 - 1 \right) \\ &= \frac{n}{2(1-\theta)} \left(\left[\varrho\left(\frac{\Psi(v)}{n}\right) \right]^2 - (1-\theta) \right) \\ &\leq \frac{n}{2(1-\theta)} \left(\left[1 + \sqrt{2\frac{\Psi(v)}{n}} \right]^2 - (1-\theta) \right) \\ &= \frac{n}{2(1-\theta)} \left(\left[1 + 2\frac{\Psi(v)}{n} + 2\sqrt{2\frac{\Psi(v)}{n}} \right] - (1-\theta) \right) \\ &\leq \frac{n}{2(1-\theta)} \left(\theta + 2\frac{\tau}{n} + 2\sqrt{2\frac{\tau}{n}} \right) \\ &= \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)}. \end{aligned}$$

D'où le resultat. \square

On note par :

$$\Psi_0 = \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)} = L(n, \theta, \tau); \quad (28)$$

comme une borne supérieure de Ψ , pendant le processus de l'algorithme.

3.2 Analyse de la complexité de l'algorithme

Dans cette section, on calcule le pas de déplacement, on prouve la décroissance de la fonction barrière de proximité à chaque itération interne et on donne les résultats de

complexité de l'algorithme.

Pour μ fixé, si on prend α le pas de déplacement, alors le nouvel itéré est donné par

$$x_+ := x + \alpha\Delta x; \quad y_+ = y + \alpha\Delta y; \quad z_+ := z + \alpha\Delta z.$$

On utilise les formules (2) et (3), on obtient :

$$\begin{aligned} x_+ & : = x(e + \alpha\frac{\Delta x}{x}) = x(e + \alpha\frac{dx}{v}) = \frac{x}{v}(v + \alpha dx); \\ z_+ & : = z(e + \alpha\frac{\Delta z}{z}) = z(e + \alpha\frac{dz}{v}) = \frac{z}{v}(v + \alpha dz); \end{aligned}$$

on a :

$$v_+ = \sqrt{\frac{x_+ z_+}{\mu}} = \sqrt{(v + \alpha dx)(v + \alpha dz)}.$$

Pour tout $\alpha \geq 0$, on pose :

$$f(\alpha) = \Psi(v_+) - \Psi(v).$$

Donc $f(\alpha)$ est la différence de la proximité entre le nouvel itéré et l'ancien itéré.

D'après le Lemme 3.1.1 (a), on a :

$$\Psi(v_+) = \Psi(\sqrt{(v + \alpha dx)(v + \alpha dz)}) \leq \frac{1}{2}(\Psi(v + \alpha dx) + \Psi(v + \alpha dz)).$$

On a :

$$f(\alpha) \leq f_1(\alpha),$$

telle que :

$$f_1(\alpha) = \frac{1}{2} [\Psi(v + \alpha dx) + \Psi(v + \alpha dz)] - \Psi(v). \quad (29)$$

avec

$$f(0) = f_1(0) = 0.$$

La dérivée de f_1 au point α est :

$$f_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi'(v_i + \alpha [dx]_i) [dx]_i + \psi'(v_i + \alpha [dz]_i) [dz]_i),$$

où $[dx]_i$ et $[dz]_i$ désignent respectivement la $i^{\text{ème}}$ composante des vecteurs dx et dz .

On utilise la mesure de proximité dans (8) et (5), on déduit que :

$$f_1'(0) = \frac{1}{2} \nabla \Psi(v)^T (dx + dz) = -\frac{1}{2} \nabla \Psi(v)^T \nabla \Psi(v) = -2\delta(v)^2.$$

La dérivée de $f_1'(\alpha)$ est :

$$f_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi''(v_i + \alpha [dx]_i) [dx]_i^2 + \psi''(v_i + \alpha [dz]_i) [dz]_i^2). \quad (30)$$

On a $f_1'' > 0$ si et seulement si $dx \neq 0$ où $dz \neq 0$, alors dans ce cas f_1 est strictement convexe. On note par :

$$v_1 = \min(v), \delta := \delta(v). \Psi = \Psi(v).$$

Les prochains lemmes donnent une relation entre Ψ et δ .

Lemme 3.2.1 *Soit δ définie dans (8). Donc, on obtient :*

$$\delta(v) \geq \sqrt{\frac{1}{2} \Psi(v)}.$$

Preuve 10 *On utilise le Lemme 3.1.4 (23), on obtient :*

$$\Psi(v) = \sum_{i=1}^n \psi(v_i) \leq \sum_{i=1}^n \frac{1}{2} (\psi'(v_i))^2 = \frac{1}{2} \|\nabla \Psi(v)\|^2 = 2\delta(v),$$

donc

$$\delta(v) \geq \sqrt{\frac{1}{2} \Psi(v)}.$$

D'où le résultat. \square

Corollaire 3.2.1 Dans la suite, on suppose que $\Psi(v) \geq \tau \geq 1$. D'après le lemme précédent on a :

$$\delta(v) \geq \frac{1}{\sqrt{2}}.$$

On utilise les Lemmes 4.1– 4.3 dans [12], nous avons les lemmes suivants.

Lemme 3.2.2 Soit f_1 définie dans (29) et δ dans (8), alors on a :

$$f_1''(\alpha) \leq 2\delta^2\psi''(v_1 - 2\alpha\delta).$$

Lemme 3.2.3 Si le pas de déplacement α vérifie l'inégalité suivante :

$$-\psi'(v_1 - 2\alpha\delta) + \psi'(v_1) \leq 2\delta, \quad (31)$$

alors :

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

Preuve 11 On a :

$$\begin{aligned} f_1'(\alpha) &= f_1'(0) + \int_0^\alpha f_1''(\zeta)d\zeta \\ &\leq -2\delta^2 + 2\delta^2 \int_0^\alpha \psi''(v_1 - 2\zeta\delta)d\zeta \\ &= -2\delta^2 - \delta \int_0^\alpha \psi''(v_1 - 2\zeta\delta)d(v_1 - 2\zeta\delta) \\ &= -2\delta^2 - \delta(\psi'(v_1 - 2\alpha\delta) - \psi'(v_1)). \end{aligned}$$

Alors, $f_1'(\alpha) \leq 0$, si

$$-\psi'(v_1 - 2\alpha\delta) + \psi'(v_1) \leq 2\delta.$$

D'où le résultat. \square

Lemme 3.2.4 *L'estimation de la plus grande valeur de α vérifiant (31) est donnée par :*

$$\bar{\alpha} = \frac{1}{2\delta}(\rho(\delta) - \rho(2\delta)). \quad (32)$$

Lemme 3.2.5 *Soient ρ et $\bar{\alpha}$ (définie dans le Lemme 3.2.4), alors on a :*

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

Preuve 12 *Par la définition de ρ on a :*

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

on dérive par rapport à δ , on trouve

$$\frac{-\psi''(\rho(\delta))\rho'(\delta)}{2} = 1,$$

alors

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

Donc ρ est décroissante par rapport à δ . Utilisant (32), on trouve :

$$\bar{\alpha} = \frac{1}{2\delta} \int_{2\delta}^{\delta} \rho'(\sigma) d\sigma = \frac{1}{\delta} \int_{\delta}^{2\delta} \frac{d\sigma}{\psi''(\rho(\sigma))}.$$

Pour obtenir une borne inférieure de $\bar{\alpha}$, il faut que $\psi''(\rho(\sigma))$ soit maximale, pour $\sigma \in \delta, 2\delta$. De (22), ψ'' est strictement décroissante. Donc, $\psi''(\rho(\sigma))$ est maximale lorsque $\rho(\sigma)$ est minimale pour $\sigma \in [\delta, 2\delta]$. De plus la fonction ρ est strictement décroissante, cela se produit lorsque $\sigma = 2\delta$. Donc

$$\bar{\alpha} = \frac{1}{\delta} \int_{\delta}^{2\delta} \frac{d\sigma}{\psi''(\rho(\sigma))} \geq \frac{\delta}{\delta \psi''(\rho(2\delta))} = \frac{1}{\psi''(\rho(2\delta))},$$

ce qui prouve le lemme. \square

Lemme 3.2.6 Soit $\bar{\alpha}$ définie dans (32). Si $\Psi(v) \geq \tau \geq 1$, on a :

$$\bar{\alpha} \geq \frac{1}{1 + (2q + 1)(1 + 4\delta) [\log(2 + 8\delta) + 1]^{\frac{q+1}{q}}}.$$

Preuve 13 Utilisant les Lemmes 3.1.6- 3.2.1- 3.2.5 et (22), on a :

$$\bar{\alpha} \geq \frac{1}{\psi''(\rho(2\delta))} \geq \frac{1}{\psi''(\rho(1 + 4\delta))},$$

on pose $t = \rho(1 + 4\delta)$, ($0 < t \leq 1$), il s'ensuit que

$$\begin{aligned} \bar{\alpha} &\geq \frac{1}{\psi''(t)} = \frac{1}{1 + \frac{1}{2t^2} + \left[\frac{1}{2}(q+1)t^{-(q+2)} + \frac{1}{2}qt^{-(2q+2)}\right] e^{t^{-q}-1}} \\ &= \frac{1}{1 + \frac{1}{2t^2} + [(q+1)t^{-1} + qt^{-(q+1)}] \left(-\psi'_b(t) - \frac{1}{2t}\right)} \\ &= \frac{1}{1 - \frac{q}{2t^2}(1 + t^{-q}) + [(q+1)t^{-1} + qt^{-(q+1)}] \left(-\psi'_b(t)\right)} \\ &> \frac{1}{1 + (2q+1)t^{-(q+1)} \left(-\psi'_b(t)\right)} \\ &> \frac{1}{1 + (2q+1)(1+4\delta) [\log(2+8\delta) + 1]^{\frac{q+1}{q}}}. \end{aligned}$$

Ce qui achève la démonstration. \square

On pose :

$$\tilde{\alpha} = \frac{1}{1 + (2q + 1)(1 + 4\delta) [\log(2 + 8\delta) + 1]^{\frac{q+1}{q}}}. \quad (33)$$

$\tilde{\alpha}$ est appelée, le pas par défaut.

Lemme 3.2.7 (Lemme 1.3.3 : [46]) On suppose que h est une fonction convexe et deux fois différentiable, telle que :

$$h(0) = 0, h'(0) < 0,$$

et h atteint son minimum global à $t^* > 0$, et h'' est croissante pour tout $t \in [0, t^*]$, alors,

$$h(t) \leq \frac{th'(0)}{2}, \text{ pour tout } t \in [0, t^*].$$

Lemme 3.2.8 Si le pas de déplacement α vérifie $\alpha \leq \bar{\alpha}$, on a :

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

Preuve 14 Considérons la fonction h définie par :

$$h(\alpha) = -2\alpha\delta^2 + \alpha\delta\psi'(v_1) - \frac{1}{2}\psi(v_1) + \frac{1}{2}\psi(v_1 - 2\alpha\delta),$$

alors

$$h(0) = f_1(0) = 0, h'(0) = f_1'(0) = -2\delta^2 \text{ et } h''(\alpha) = 2\delta^2\psi''(v_1 - 2\alpha\delta) > 0,$$

d'après le Lemme 3.2.2 on a :

$$f_1''(\alpha) \leq h''(\alpha),$$

et par conséquent on obtient

$$f_1'(\alpha) \leq h'(\alpha) \text{ et } f_1(\alpha) \leq h(\alpha),$$

donc si $\alpha \leq \tilde{\alpha}$ on a :

$$\begin{aligned}
h'(\alpha) &= h'(0) + \int_0^\alpha h''(\zeta) d\zeta, \\
&= -2\delta^2 + 2\delta^2 \int_0^\alpha \psi''(v_1 - 2\zeta\delta) d\zeta, \\
&= -2\delta^2 - \delta \int_0^\alpha \psi''(v_1 - 2\zeta\delta) d(v_1 - 2\zeta\delta) \\
&= -2\delta^2 + \delta(\psi'(v_1 - 2\alpha\delta) + \psi'(v_1)), \\
&\leq -2\delta^2 + 2\delta^2 = 0,
\end{aligned}$$

avec h'' est croissante, donc d'après le Lemme 3.2.7 on a :

$$f(\alpha) \leq f_1'(\alpha) \leq h(\alpha) \leq \frac{\alpha h'(0)}{2} = -\alpha\delta^2.$$

D'où le résultat. \square

Lemme 3.2.9 Soit $\tilde{\alpha}$ le pas défini dans (33) et soit $\Psi(v) \geq \tau \geq 1$. Alors

$$f(\tilde{\alpha}) \leq -\frac{\sqrt{\Psi(v)}}{2 + (2q + 1)(1 + 4\sqrt{2}) [\log(2 + 4\sqrt{2\Psi_0}) + 1]^{\frac{q+1}{q}}}. \quad (34)$$

Preuve 15 En combinant les résultats des Lemmes 3.2.6 et 3.2.8, on obtient :

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

Du fait que $\tilde{\alpha} < \bar{\alpha}$, on a

$$\begin{aligned}
f(\tilde{\alpha}) &\leq -\frac{\delta^2}{1 + (2q + 1)(1 + 4\delta) [\log(2 + 8\delta) + 1]^{\frac{q+1}{q}}} \\
&\leq -\frac{\delta}{\sqrt{2} + (2q + 1)(\sqrt{2} + 4) [\log(2 + 8\delta) + 1]^{\frac{q+1}{q}}}.
\end{aligned}$$

Puisque la diminution dépend de façon monotone de δ , la substitution rend

$$\begin{aligned} f(\tilde{\alpha}) &\leq -\frac{\sqrt{\Psi(v)}}{2 + (2q + 1)(2 + 4\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi(v)}) + 1 \right]^{\frac{q+1}{q}}} \\ &\leq -\frac{\sqrt{\Psi(v)}}{2 + (2q + 1)(2 + 4\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}}}. \end{aligned}$$

Où la dernière inégalité découle de $\Psi_0 \geq \Psi \geq \tau \geq 1$.

Ce qui achève la démonstration. \square

On note par Ψ_0 la première μ -mise à jour de $\Psi(v)$ et , $\Psi_k, k = 1, 2, \dots, K - 1$ suite des valeurs de $\Psi(v)$ et K désigne le nombre total d'itérations internes dans l'itération externe, de plus nous avons

$$\Psi_{k+1} \leq \Psi_k - \frac{(\Psi_k)^{\frac{1}{2}}}{2 + (2q + 1)(2 + 4\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}}}$$

Utilisant le Lemme 2.1.4 avec

$$t_k = \Psi_k, \beta = \frac{1}{2 + (2q + 1)(2 + 4\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}}} \text{ et } \gamma = \frac{1}{2},$$

on déduit le résultat suivant :

Lemme 3.2.10 Soit K le nombre total d'itérations internes dans l'itération externe.

Alors,

$$K \leq \left(4 + (2q + 1)(4 + 8\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}} \right) \Psi_0^{\frac{1}{2}}. \quad (35)$$

Preuve 16 Du Lemme 2.1.4, on obtient

$$K \leq \frac{\Psi_0^\gamma}{\beta\gamma} = \left(4 + (2q + 1)(4 + 8\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}} \right) \Psi_0^{\frac{1}{2}}.$$

D'où le résultat. \square

Maintenant, nous estimons le nombre total d'itérations produit par l'algorithme 4.

Théorème 3.2.1 *Si $\tau \geq 1$, le nombre total d'itérations est au plus*

$$\left(4 + (2q + 1)(4 + 8\sqrt{2}) \left[\log(2 + 4\sqrt{2\Psi_0}) + 1\right]^{\frac{q+1}{q}}\right) \Psi_0^{\frac{1}{2}} \frac{\log \frac{n}{\varepsilon}}{\theta}. \quad (36)$$

Preuve 17 *Il suffit d'appliquer le Théorème 2.1.2 et (10). \square*

Nous obtenons pour les méthodes à grand pas, avec $\tau = \mathcal{O}(\sqrt{n})$ et $\theta = \Theta(1)$, une borne de $\mathcal{O}(q\sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log \frac{n}{\varepsilon})$ itérations où $\Psi_0 = \mathcal{O}(n)$. En particulier, si $q = 2$, on obtient la complexité suivante :

$$\mathcal{O}(\sqrt{n}(\log n)^{\frac{3}{2}} \log \frac{n}{\varepsilon}).$$

Dans le cas des méthodes à petit pas, la meilleure complexité est obtenue comme suit.

Par (24), avec

$$\psi(t) \leq \frac{2+q}{2}(t-1)^2, \quad t > 1,$$

on a :

$$\begin{aligned} \Psi(v_+) &\leq n\psi\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right)\right) \\ &\leq \frac{n(2+q)}{2} \left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right) - 1\right)^2 \\ &= \frac{n(2+q)}{2(1-\theta)} \left(\varrho\left(\frac{\Psi(v)}{n}\right) - \sqrt{1-\theta}\right)^2. \end{aligned}$$

Utilisant (25) et $\Psi(v) \leq \tau$, on obtient

$$\begin{aligned}
\frac{n(2+q)}{2(1-\theta)} \left(\varrho\left(\frac{\Psi(v)}{n}\right) - \sqrt{1-\theta} \right)^2 &\leq \frac{n(2+q)}{2(1-\theta)} \left(1 + \sqrt{\frac{2\Psi(v)}{n}} - \sqrt{1-\theta} \right)^2 \\
&= \frac{n(2+q)}{2(1-\theta)} \left(1 - \sqrt{1-\theta} + \sqrt{\frac{2\Psi(v)}{n}} \right)^2 \\
&\leq \frac{n(2+q)}{2(1-\theta)} \left(\theta + \sqrt{\frac{2\Psi(v)}{n}} \right)^2 \\
&\leq \frac{n(2+q)}{2(1-\theta)} \left(\theta + \sqrt{\frac{2\tau}{n}} \right)^2 \\
&= \frac{2+q}{2(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau} \right)^2 = \Psi_0.
\end{aligned}$$

On a $\tau = \mathcal{O}(1)$ et $\theta = \Theta(\frac{1}{\sqrt{n}})$, dans ce cas $\Psi_0 = \mathcal{O}(q)$ et une borne de

$$\mathcal{O}\left(q^{\frac{3}{2}} (\log \sqrt{q})^{\frac{q+1}{q}} \sqrt{n} \log \frac{n}{\varepsilon}\right),$$

itérations.

3.3 Expérimentations numériques

Le but de cette section est d'examiner l'influence du choix de la nouvelle fonction noyau sur le comportement numérique de l'algorithme primal-dual pour résoudre un programme linéaire donné dans l'Algorithme 3. L'algorithme est codé dans MATLAB (R2014a), effectué sur PC avec Processeur Authentique Intel (R) CPR T2080 @ 1,73GHZ et mémoire installée (RAM) 2,00GO. Pour les paramètres τ , θ et le paramètre de précision ϵ , nous avons fixé ces paramètres à $\tau = \sqrt{n}$, $\theta \in \{0.3, 0.5, 0.7, 0.9, 0.99\}$ et $\epsilon = 10^{-4}$. Le choix du pas α , est un autre problème crucial dans l'analyse de l'algorithme. Il doit être conçu de telle sorte que la proximité des itérés par rapport au centre actuel s'améliore suffisamment. Dans le cas théorique, on attribue généralement une valeur α très petite

à chaque itération interne. En pratique, cela conduit à un très grand nombre d'itérations internes. Donc, pour accélérer le processus des itérations, nous proposons un choix dynamique et pratique défini ci-dessous :

Proposition 4 *On suppose que $(x, y, z) \in \mathcal{F}_{(P)} \times \mathcal{F}_{(D)}$, $(\Delta x, \Delta y, \Delta z)$ satisfait le système (1) et le triplet $(x + \Delta x, y + \Delta y, z + \Delta z) \in \mathcal{F}_{(P)} \times \mathcal{F}_{(D)}$, si $\|\Delta x\| \leq \delta$ ($\delta > 0$), alors il existe toujours un $p > 0$, tel que $(x + \alpha p \Delta x, y + \alpha p \Delta y, z + \alpha p \Delta z) \in \mathcal{F}_{(P)} \times \mathcal{F}_{(D)}$.*

Preuve 18 *Voir (Proposition 4.1 [50]).*

Choix dynamique

D'après la proposition précédente, nous agrandissons la taille du pas de déplacement, en utilisant la procédure suivante :

Nous prenons $\alpha = p\tilde{\alpha}$, où $p \geq 1$ est un scalaire fixe dépendant de Δx ou Δz et $\tilde{\alpha}$ est le pas par défaut (choix théorique).

Dans nos tests numériques, on pose :

$$p = \begin{cases} p_1 & \text{si} & \|\Delta x\| \leq 1 \\ p_2 & \text{si} & 1 \leq \|\Delta x\| \leq n \\ p_3 & \text{si} & \|\Delta x\| \geq n \end{cases} ,$$

où p_1, p_2, p_3 sont des scalaires positifs, bien choisis.

Choix pratique : On utilise la procédure de positivité (page 32).

Exemple 5 *Soit le programme linéaire suivant : $m = 3, n = 6$,*

$$A = \begin{pmatrix} 2 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} ,$$

$$c = \left(3, -1, 1, 0, 0, 0 \right)^T \text{ et } b = \left(0, 0, 1 \right)^T .$$

Le point de départ est :

$$x_0 = [0.06757, 0.13258, 0.13302, 0.26774, 0.13302, 0.2664]^T, z_0 = [10, 4, 6, 1, 5, 1]^T,$$

$$y_0 = [-2, -2, -3]^T.$$

La solution optimale est :

$$x^* = (0.0000, 0.5000, 0.0000, 0.5000, 0.0000, 0.0004)^T,$$

$$y^* = (-0.5000, -0.4902, -0.5000)^T,$$

$$z^* = (4.5000, 0.0000, 1.9902, 0.0000, 0.9902, 0.0098)^T.$$

Dans les tableaux des résultats, n représente la taille de l'exemple, (Ext) représente le nombre d'itérations externes, (Int) représente le nombre d'itérations internes et (Temps) représente le temps de calcul en secondes. Le tableau suivant donne le nombre d'itérations pour les différentes valeurs de θ et p . Le cas $\theta = 0,9$ donne le nombre d'itérations le plus bas dans tous les cas.

θ	p_1	p_2	p_3	Int	Ext	Temps
0.3	200	100	50	1	31	0.027893
0.5	201	100	50	1	16	0.029697
0.7	201	100	50	2	10	0.032418
0.9	423	100	50	2	5	0.019141
0.99	422	100	50	13	3	0.074557

Tableau1 : Nombre d'itérations internes et externes pour plusieurs choix de θ et p .

Choix	Int	Ext	Temps
Dynamique	$p_1 = 423 \quad p_2 = 100 \quad p_3 = 50$	5	0.0191410
	2 $p_1 = 100 \quad p_2 = 50 \quad p_3 = 25$ 25		0.1593165
Théorique	2174	5	7.715739
Pratique	4	5	0.024999

Tableau2 : Nombre d'itérations internes et externes pour plusieurs choix de α
et $\theta = 0,9$.

Exemple 6 On considère le programme linéaire suivant : $m = 5, n = 9$,

$$A = \begin{pmatrix} 0 & 1 & 2 & -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 1 & 2 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & -1 & -2 & 0 & 0 & 0 & 1 \\ 1 & 3 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, c = \left(1, 0, -2, 1, 1, 0, 0, 0, 0 \right)^T$$

et $b = \left(1, 2, 3, 2, 1 \right)^T$.

Le point de départ est :

$$x_0 = [0.1819, 0.0699, 0.063, 0.1105, 0.2012, 0.6732, 1.1885, 2.835, 2.1912]^T,$$

$$z_0 = [4.939, 3.544, 4.7186, 9.1788, 4.5072, 1.384, 0.875, 0.4241, 0.4463]^T,$$

$$y_0 = [-1.3843, -0.8751, -0.4241, -0.4463, -3.0424]^T.$$

La solution optimale est :

$$x^* = (0, 0, 0.2664, 0, 0, 0.4269, 1.1406, 3.5729, 2)^T,$$

$$y^* = \left(0, 0, 0, 0, -0.4999 \right)^T \text{ et}$$

$$z^* = \left(1.5, 1.4999, 0, 1.9999, 1.4999, 0, 0, 0, 0 \right)^T.$$

choix	Int	Ext	Temps
Théorique	2704	5	24.627917
Dynamique <small>$p_1=100, p_2=50$ et $p_3=25$</small>	23	5	0.218753
Pratique	4	5	0.109279

Tableau3 : Nombre d'itérations internes et externes pour plusieurs choix de α et $\theta = 0, 9$.

Exemple 7 Nous considérons l'exemple suivant : $n = 2m$

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

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

Le point de départ :

$$x_0(i) = x_0(i + m) = 1, \quad z_0(i) = 1, \quad z_0(i + m) = 2 \text{ et } y_0(i) = -2 \text{ pour } i = 1, \dots, m.$$

La solution optimale est :

$$x^* = \begin{cases} 2 & \text{si } i = 1, \dots, m \\ 0 & \text{si } i = m + 1, \dots, n \end{cases}, \quad y^* = -1, \text{ pour } i = 1, \dots, m$$

et

$$z^* = \begin{cases} 0 & \text{si } i = 1, \dots, m \\ 1 & \text{si } i = m + 1, \dots, n. \end{cases}$$

Nous avons les résultats ci-dessous :

n	P			<i>Théorique</i>		<i>Dynamique</i>		<i>Pratique</i>	
	p_1	p_2	p_2	<i>Int</i>	<i>Temps</i>	<i>Int</i>	<i>Temps</i>	<i>Int</i>	<i>Temps</i>
20	500	350	150	4171	484.938590	4	0.208902	4	0.122828
50	1050	350	150	6977	791.156169	3	0.491385	4	0.279962
100	1050	350	150	10385	3475.590158	4	2.937549	4	0.360571
200	2000	350	280	15547	19737.654690	5	12.924049	5	4.378243
400	3010	500	280	4	68.116663	5	21.679448
500	3110	510	280	5	137.160435	4	42.884622
1000	5525	510	350	6	1321.177117	5	262.051496

Tableau4 : Nombre d'itérations internes pour plusieurs choix de α et $\theta = 0, 9$.

Conclusion 5 *En pratique, le choix du pas de déplacement joue un rôle crucial dans le comportement numérique de l'algorithme. Pour accélérer le processus d'itération de notre algorithme, nous avons proposé deux choix : dynamique et pratique. L'algorithme avec le choix pratique fonctionne plus vite que celui du dynamique, mais pour des valeurs appropriées du paramètre p , les deux choix conduisent à une diminution considérable du nombre total d'itérations, par rapport au choix théorique.*

Deuxième partie

Extension au problème semi-défini linéaire

Dans cette étape, on généralise l'étude faite dans la première partie voir [23], au cas semi-défini. On obtient la même complexité qu'au cas linéaire c'est-à-dire :

$$\mathcal{O}(q\sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log \frac{n}{\epsilon})$$

pour les méthodes à grand pas et

$$\mathcal{O}(q^{\frac{3}{2}}(\log \sqrt{q})^{\frac{q+1}{q}} \sqrt{n} \log \frac{n}{\epsilon})$$

pour les méthodes à petit pas. De plus, on effectue quelques tests numériques pour montrer l'efficacité de cette approche et comparer nos résultats à ceux obtenus avec les méthodes basées sur différentes fonctions noyaux. On considère le système (15) et la fonction de proximité $\Psi(V)$ définie dans (16), où $\psi(t)$ est définie dans (21).

3.4 Description de l'algorithme

L'algorithme 4 proposé dans le chapitre précédent se déroule comme suit : Supposons que l'on donne un point strictement réalisable (X, y, Z) qui est dans un τ -voisinage (c-à-d $\Psi(V) < \tau$) de μ -centre, avec $\tau \geq 1$ la valeur seuil. Si $\Psi(V) < \tau$, on commence une nouvelle itération externe en effectuant une mise à jour du paramètre μ via : $\mu^+ = (1 - \theta)\mu$, pour $0 < \theta < 1$ puis on résout le système de Newton (17) pour obtenir la direction de descente. La condition de positivité d'une nouvelle itération est assurée avec le bon choix du pas de déplacement α . Cette procédure est répétée jusqu'à ce qu'on trouve un nouvel itéré (X^+, y^+, Z^+) qui est dans un τ -voisinage de μ^+ -centre. Ce processus se répète jusqu'à ce que $n\mu \leq \epsilon$.

Les paramètres τ , θ et α décrits dans l'algorithme sont choisis de telle manière que le nombre d'itérations soit minimal.

Par la suite on montre l'influence de la mise à jour de μ sur la valeur de la fonction barrière $\Psi(X, y, Z) := \Psi(V)$ après chaque itération externe.

3.4.1 Borne supérieure de $\Psi(V)$ pour chaque itération externe

Au début de chaque itération externe de l'algorithme, juste avant la mise à jour du paramètre μ , la matrice V est mise à jour par la relation suivante : $V^+ = \frac{V}{\sqrt{1-\theta}}$. Ce qui conduit en général à une augmentation de la valeur de $\Psi(V)$ dans les itérations externes, puis, au cours des itérations internes $\Psi(V)$ diminue jusqu'à ce qu'elle dépasse à nouveau le seuil τ . Pour cela, on trouve une borne supérieure de $\Psi(V^+)$.

Le prochain théorème est une extension du Théorème 3.2 [12] aux matrices définies positives.

Théorème 3.4.1 (Théorème 3.3.2 [29]) *Soit ϱ la fonction inverse définie dans le Lemme 3.1.5. Alors pour toute matrice positive V et pour tout $\beta \geq 1$, on a :*

$$\Psi(\beta V) \leq n\psi \left(\beta \varrho \left(\frac{\Psi(V)}{n} \right) \right).$$

Corollaire 3.4.1 *Soit $0 \leq \theta < 1$ et $V_+ = \frac{V}{\sqrt{1-\theta}}$. Si $\Psi(V) \leq \tau$, alors*

$$\Psi(V_+) \leq n\psi \left(\frac{\varrho \left(\frac{\tau}{n} \right)}{\sqrt{1-\theta}} \right) \leq \frac{n\theta + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)} = \Psi_0.$$

Preuve 19 *Du fait que $\frac{1}{\sqrt{1-\theta}} \geq 1$ et $\varrho \left(\frac{\Psi(V)}{n} \right) \geq 1$, on a $\frac{1}{\sqrt{1-\theta}} \varrho \left(\frac{\Psi(V)}{n} \right) \geq 1$. Utilisant le théorème précédent avec $\beta = \frac{1}{\sqrt{1-\theta}}$ et la croissance de ϱ , on trouve*

$$\Psi(V_+) \leq n\psi \left(\frac{\varrho \left(\frac{\Psi(V)}{n} \right)}{\sqrt{1-\theta}} \right) \leq n\psi \left(\frac{\varrho \left(\frac{\tau}{n} \right)}{\sqrt{1-\theta}} \right).$$

Ceci prouve la première inégalité.

Pour la deuxième inégalité, on procède à la même démonstration du Lemme 3.1.8. \square

3.4.2 Décroissance de $\Psi(V)$ durant une itération interne

Dans chaque itération interne, les directions de recherche ΔX , Δy et ΔZ sont obtenues en résolvant le système (17). Le nouvel itéré est donné par :

$$X_+ = X + \alpha\Delta X, \quad y_+ = y + \alpha\Delta y, \quad Z_+ = Z + \alpha\Delta Z, \quad \alpha > 0.$$

En utilisant (14), nous pouvons écrire

$$X_+ = X + \alpha\Delta X = X + \alpha\sqrt{\mu}DD_XD = \sqrt{\mu}D(V + \alpha D_X)D,$$

et

$$Z_+ = Z + \alpha\Delta Z = Z + \alpha\sqrt{\mu}D^{-1}D_ZD^{-1} = \sqrt{\mu}D^{-1}(V + \alpha D_Z)D^{-1}.$$

Dénotant par V_+ , la matrice V après un pas α , alors

$$V_+ = \frac{1}{\sqrt{\mu}} (D^{-1}X_+Z_+D)^{\frac{1}{2}}.$$

Donc

$$V_+^2 = (V + \alpha D_X)(V + \alpha D_Z),$$

et

$$V_+^2 = (V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_Z)(V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_X)^{-\frac{1}{2}}.$$

On suppose que $V + \alpha D_X \succ 0$ et $V + \alpha D_Z \succ 0$ pour le pas α qui assure la stricte faisabilité. Alors il est clair que V_+^2 est similaire à la matrice suivante :

$$(V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_Z)(V + \alpha D_X)^{\frac{1}{2}}$$

Et par conséquent, on déduit que les valeurs propres de V_+ sont les mêmes que la matrice

$$\tilde{V}_+ = \left((V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_Z)(V + \alpha D_X)^{\frac{1}{2}} \right)^{\frac{1}{2}}.$$

Par la définition de $\Psi(V)$, on obtient

$$\Psi(V_+) = \Psi(\tilde{V}_+).$$

Notre objectif est de trouver α telle que :

$$f(\alpha) = \Psi(V_+) - \Psi(V) = \Psi(\tilde{V}_+) - \Psi(V),$$

soit minimale.

Lemme 3.4.1 (*Proposition 5.2.6 [43]*) *Pour tout $V_1, V_2 \succ 0$, alors*

$$\Psi \left(\left[V_1^{\frac{1}{2}} V_2 V_1^{\frac{1}{2}} \right]^{\frac{1}{2}} \right) \leq \frac{1}{2} (\Psi(V_1) + \Psi(V_2)).$$

En utilisant le Lemme 3.4.1, on obtient :

$$\begin{aligned} \Psi(\tilde{V}_+) &= \Psi \left(\left((V + \alpha D_X)^{\frac{1}{2}} (V + \alpha D_Z) (V + \alpha D_X)^{\frac{1}{2}} \right)^{\frac{1}{2}} \right) \\ &\leq \frac{1}{2} (\Psi(V + \alpha D_X) + \Psi(V + \alpha D_Z)). \end{aligned}$$

On a $f(\alpha) \leq f_1(\alpha)$, où

$$f_1(\alpha) = \frac{1}{2} (\Psi(V + \alpha D_X) + \Psi(V + \alpha D_Z)) - \Psi(V),$$

est une fonction convexe par rapport à α car Ψ est convexe. Evidemment, $f(0) = f_1(0) = 0$. Maintenant, pour estimer la décroissance de la fonction de proximité durant un pas α , on utilise les deux dérivées successives de f_1 par rapport à α . En utilisant la règle de différentiabilité pour les fonctions matricielles, on obtient

$$f_1'(\alpha) = \frac{1}{2} \text{tr} (\psi'(V + \alpha D_X) D_X + \psi'(V + \alpha D_Z) D_Z),$$

et

$$f_1''(\alpha) = \frac{1}{2} \text{tr} (\psi''(V + \alpha D_X) D_X^2 + \psi''(V + \alpha D_Z) D_Z^2). \quad (36)$$

Par conséquent, en utilisant (20) et la dernière équation de (17), on obtient

$$f_1'(0) = \frac{1}{2} \text{tr} (\psi'(V) (D_X + D_S)) = \frac{1}{2} \text{tr} (-\psi'(V)^2) = -2\delta(V)^2.$$

En partant de (36) et en utilisant les mêmes arguments que dans [51], on obtient le lemme suivant

Lemme 3.4.2 (*Lemme 5.19 dans [54]*). *On a :*

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

Posant $v_1 = \lambda_n(V), v_2 = \lambda_{n-1}(V), \dots, v_n = \lambda_1(V)$, on obtient

$$f_1''(\alpha) \leq 2\delta^2 \psi''(v_1 - 2\alpha\delta),$$

qui est la même inégalité que dans le Lemme 3.2.2. A partir de ce stade, nous pouvons appliquer mot à mot les mêmes arguments que dans [12] pour le cas (*PL*) afin d'obtenir les résultats suivants.

$$- f(\tilde{\alpha}) \leq - \frac{\sqrt{\Psi(v)}}{2 + (2q+1)(2+4\sqrt{2}) \left[\log(2+4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}}}.$$

– Le nombre total des itérations internes dans les itérations externes est donné par :

$$K \leq \left(4 + (2q+1)(4+8\sqrt{2}) \left[\log(2+4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}} \right) \Psi_0^{\frac{1}{2}}.$$

– La borne supérieure pour le nombre total des itérations est :

$$\left(4 + (2q+1)(4+8\sqrt{2}) \left[\log(2+4\sqrt{2\Psi_0}) + 1 \right]^{\frac{q+1}{q}} \right) \Psi_0^{\frac{1}{2}} \frac{\log \frac{n}{\varepsilon}}{\theta}.$$

3.5 Expérimentations numériques

Dans cette section, nous présentons quelques résultats numériques. On considère le problème (*SDP*) présenté dans [54], avec les données suivantes :

Exemple 8

$$C = \begin{pmatrix} 3 & 3 & -3 & 1 & 1 \\ 3 & 5 & 3 & 1 & 2 \\ -3 & 3 & -1 & 1 & 2 \\ 1 & 1 & 1 & -3 & -1 \\ 1 & 2 & 2 & -1 & -1 \end{pmatrix}, A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & -1 \\ 0 & -1 & 1 & -1 & -2 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & -2 & 2 & 0 \\ 0 & 2 & 1 & 0 & 2 \\ -2 & 1 & -2 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 2 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 2 & 2 & -1 & -1 & 1 \\ 2 & 0 & 2 & 1 & 1 \\ -1 & 2 & 0 & 1 & 0 \\ -1 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & -2 \end{pmatrix}, b = \begin{pmatrix} -2 \\ 2 \\ -2 \end{pmatrix}.$$

On peut facilement vérifier que $X = I$ est une solution réalisable pour le problème primal, et que $y = [1, 1, 1]^T$ et $Z = I$ le sont pour le problème dual. La solution optimale du problème primal est donnée par

$$X^* = \begin{pmatrix} 0.0714 & -0.0718 & 0.0169 & 0.0649 & -0.1583 \\ -0.0718 & 0.0724 & -0.0183 & -0.0602 & 0.1676 \\ 0.0169 & -0.0183 & 0.0103 & -0.0084 & -0.0772 \\ 0.0649 & -0.0602 & -0.0084 & 0.1481 & 0.0056 \\ -0.1583 & 0.1676 & -0.0772 & 0.0056 & 0.6022 \end{pmatrix},$$

et la solution optimale du problème dual est donnée par

$$Z^* = \begin{pmatrix} 1.4338 & 0.5754 & -0.0295 & -0.4043 & 0.2169 \\ 0.5754 & 1.0956 & 0.3401 & 0.2169 & -0.1120 \\ -0.0295 & 0.3401 & 1.1874 & 0.2169 & 0.0478 \\ -0.4043 & 0.2169 & 0.2169 & 0.2831 & -0.1415 \\ 0.2169 & -0.1120 & 0.0478 & -0.1415 & 0.0957 \end{pmatrix}$$

$$y^* = \begin{pmatrix} 0.8585 \\ 1.0937 \\ 0.7831 \end{pmatrix}.$$

La valeur optimale des deux problèmes est égale à -1.0957 .

On considère les fonctions noyaux suivantes :

$$\psi(t) = \frac{t^2 - 1 - \log t}{2} + \frac{e^{\frac{1}{t^q} - 1} - 1}{2q} \text{ pour } t > 0, q \geq 1.$$

$$\psi_1(t) = \frac{t^2 - 1}{2} - \log t.$$

$$\psi_2(t) = \frac{t^2 - 1}{2} - (t - 1)e^{\frac{1}{t} - 1}.$$

$$\psi_3(t) = (m + 1)t^2 - (m + 2)t + \frac{1}{t^m}, t > 0 \quad \text{où } m > 4.$$

$\psi_1(t)$ est la fonction noyau logarithmique classique (Voir [51]), $\psi_2(t)$ (Voir [57]) et $\psi_3(t)$ (Voir [41]), sont des fonctions noyaux non auto-régulières. On prend le paramètre barrière de la mise à jour $\theta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, le pas de déplacement $\alpha \in \{0.5, 0.6, 0.7, \alpha_{\max}\}$.

Le pas pratique α_{\max} tels que $\alpha_{\max} = \rho \min(\alpha_X, \alpha_Z)$ et $\rho \in (0, 1)$, où

$$\alpha_X = \begin{cases} -\frac{1}{\lambda_{\min}(X^{-1}\Delta X)} & \text{si } \lambda_{\min}(X^{-1}\Delta X) < 0 \\ 1 & \text{si } \lambda_{\min}(X^{-1}\Delta X) \geq 0 \end{cases},$$

et

$$\alpha_Z = \begin{cases} -\frac{1}{\lambda_{\min}(Z^{-1}\Delta Z)} & \text{si } \lambda_{\min}(Z^{-1}\Delta Z) < 0 \\ 1 & \text{si } \lambda_{\min}(Z^{-1}\Delta Z) \geq 0 \end{cases}.$$

Le paramètre seuil $\tau = 3$ et le paramètre de précision $\epsilon = 10^{-8}$, dans tous les cas. Le nombre des itérations et le temps de calcul de l'algorithme basé sur les fonctions noyaux ci-dessus sont donnés dans les tableaux suivants :

$\theta \backslash \alpha$	0.5		0.6		0.7		α_{\max}	
	Int	Temps	Int	Temps	Int	Temps	Int	Temps
0.1	42	0.4532	33	0.3095	26	0.4330	15	0.3103
0.1	42	0.5413	33	0.3734	28	0.3685	15	0.2925
0.3	39	0.2695	31	0.2231	25	0.2083	15	0.1737
0.3	40	0.2651	32	0.2460	25	0.2147	15	0.2178
0.5	37	0.2804	26	0.1961	26	0.1286	15	0.1381
0.5	39	0.2509	27	0.2002	26	0.1441	15	0.1427
0.7	35	0.1456	28	0.2336	22	0.2672	16	0.1373
0.7	36	0.3283	31	0.2032	23	0.1719	16	0.1336
0.9	31	0.1133	23	0.1592	19	0.1446	15	0.1796
0.9	31	0.1835	23	0.1616	19	0.1577	15	0.1185

Tableau1 : Résultats numériques de $\psi(t)$ pour $q = 1, q = 2$ (alternativement)

$\theta \backslash \alpha$	0.5		0.6		0.7		α_{\max}	
	Int	Temps	Int	Temps	Int	Temps	Int	Temps
0.1	39	0.4480	31	0.3727	25	0.3483	15	0.2871
0.3	37	0.3298	30	0.2187	25	0.2083	15	0.2419
0.5	35	0.2066	26	0.2687	26	0.2462	15	0.1313
0.7	34	0.3439	27	0.0948	21	0.1918	16	0.1354
0.9	30	0.2553	23	0.2868	18	0.2298	15	0.1132

Tableau2 : Résultats numériques pour $\psi_1(t)$.

$\theta \backslash \alpha$	0.5		0.6		0.7		α_{\max}	
	Int	Temps	Int	Temps	Int	Temps	Int	Temps
0.1	41	0.3990	32	0.3406	26	0.4055	15	0.2871
0.3	38	0.3449	30	0.3135	25	0.2774	15	0.1525
0.5	36	0.3012	26	0.9552	26	0.2743	15	0.2216
0.7	34	0.1812	28	0.2142	22	0.2342	16	0.1917
0.9	31	0.1718	23	0.2442	18	0.1382	15	0.1031

Tableau3 : Résultats numériques pour $\psi_2(t)$.

$\theta \backslash \alpha$	0.5		0.6		0.7		α_{\max}	
	Int	Temps	Int	Temps	Int	Temps	Int	Temps
0.1	98	0.5745	86	0.5279	69	0.5514	69	0.5514
0.3	77	0.3306	52	0.2903	52	0.5573	48	0.4156
0.5	67	0.4727	53	0.3989	40	0.3151	29	0.3111
0.7	55	0.2651	47	0.2471	32	0.3329	52	0.2719
0.9	40	0.2812	32	0.3669	24	0.1611	52	0.3636

Tableau4 : Résultats numériques pour $\psi_3(t)$

Pour les prochains exemples, on remplace la fonction noyau ψ_3 par la fonction suivante :

$$\psi_3(t) = \frac{t^2 - 1}{2} + \frac{q^{\frac{1}{t}-1} - 1}{q \log q} - \frac{q - 1}{q} (t - 1), t > 0, q > 1, \text{ (voir [6])}$$

et $\alpha = \alpha_{\max}$

Exemple 9 Soit le problème (SDP) où

$$C = \text{diag}(-4, -5, 0, 0, 0), A_1 = \text{diag}(2, 1, 1, 0, 0),$$

$$A_2 = \text{diag}(1, 2, 0, 1, 0)$$

$$A_3 = \text{diag}(0, 1, 0, 0, 1), b = (-2, 2, -2)^T.$$

La solution initiale :

$$X^0 = \text{diag}(1, 1, 5, 4, 2),$$

$$Z^0 = \text{diag}(1, 1, 2, 1, 2),$$

$$y^0 = (-2, -1, -2)^T.$$

La solution optimale :

$$X^* = \text{diag}(3, 2, 0, 0, 1),$$

$$Z^* = \text{diag}(0, 0, 1, 2, 0),$$

$$y^* = (-1, -2, 0)^T.$$

La valeur optimale : $C \bullet X^* = -22$.

θ	$\psi(t) (q = 1)$	$\psi_1(t)$	$\psi_2(t)$	$\psi_3(t) (q = 2)$
	<i>Int Temps</i>	<i>Int Temps</i>	<i>Int Temps</i>	<i>Int Temps</i>
0.1	16 0.3098	16 0.2488	16 0.2675	16 0.2400
0.3	16 0.2346	16 0.1020	16 0.1413	16 0.2463
0.5	17 0.1438	16 0.1244	17 0.2343	17 0.1275
0.7	17 0.1297	17 0.1222	17 0.2217	17 0.2438
0.9	17 0.0649	17 0.1179	17 0.2424	17 0.2311

Tableau 5 : Comparaison des résultats numériques des quatre fonctions

Exemple 10 Soient : $C = \text{diag}(-4, -2, 0, 0, 0)$, $A_1 = \text{diag}(1, 1, 1, 0, 0)$,
 $A_2 = \text{diag}(2, -1, 0, 1, 0)$, $A_3 = \text{diag}(1, 2, 0, 0, 1)$ et $b = (4, 8, 4)$.

La solution initiale :

$$X^0 = \text{diag}(3.92, 0.02, 0.061, 0.183, 0.04),$$

$$Z^0 = \text{diag}(1, 1, 2, 1, 1),$$

$$y^0 = (-2, -1, -1)^T.$$

La solution optimale :

$$X^* = \text{diag}(4, 0, 0.001, 0.003, 0),$$

$$Z^* = \text{diag}(0, 5.99976, 0.0009, 0.0003, 3.9985),$$

$$y^* = (-0.0009, -0.0003, -3.9985)^T.$$

Avec $C \bullet X^* = b^T y^* = -16$.

θ	$\psi(q=1)$	ψ_1	ψ_2	$\psi_3(q=2)$
	Int Temps	Int Temps	Int Temps	Int Temps
0.1	17 0.3913	16 0.1921	17 0.2479	16 0.3662
0.3	17 0.0952	16 0.2336	17 0.2995	16 0.2639
0.5	17 0.2201	16 0.1251	17 0.2263	16 0.1214
0.7	17 0.1209	17 0.2259	17 0.1133	16 0.1276
0.9	15 0.1079	15 0.2438	15 0.2223	15 0.1800

Tableau 6 : Comparaison des résultats numériques des quatre fonctions

Conclusion 6

Les résultats de ces six tableaux montrent également que l'algorithme basé sur notre nouvelle fonction noyau ψ est efficace et que le nombre des itérations de l'algorithme dépend des valeurs du paramètre θ et du pas de déplacement α . En fait, pour chaque θ considéré, les plus grandes valeurs de α donnent les meilleurs nombres des itérations. Cependant, le pas α devrait avoir une limite supérieure dans le calcul pratique. Mais il n'y a pas mieux

que $\alpha = \alpha_{\max}$ qui donne les meilleurs nombres des itérations dans tous les cas.

3.6 Conclusion

Dans cette étude, on a développé dans la première partie, une méthode de points intérieurs primale-duale pour résoudre les problèmes de la programmation linéaire basée sur une nouvelle classe de fonctions noyaux paramétrisées avec un terme barrière double. On montre que l'algorithme correspondant admet une complexité polynômiale qui est de l'ordre

$$\mathcal{O}(q\sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log \frac{n}{\epsilon})$$

pour les algorithmes à grand pas et

$$\mathcal{O}(q^{\frac{3}{2}}(\log \sqrt{q})^{\frac{q+1}{q}} \sqrt{n} \log \frac{n}{\epsilon})$$

pour les algorithmes à petit pas.

En pratique, on a constaté que la valeur théorique du pas de déplacement α est très petite dans chaque itération, ce qui demande un nombre très important d'itérations et plus de temps de calcul. Pour cela, on a proposé d'autres procédures pour améliorer le comportement numérique de notre algorithme à savoir : le choix dynamique et le choix pratique qui remplacent le choix théorique. Les résultats obtenus sont en faveur avec ces deux choix. Le nombre d'itérations et le temps de calcul sont réduits d'une manière considérable.

Dans la deuxième partie, on a généralisé l'étude au cas semi-défini linéaire où on a trouvé la même complexité qu'au cas linéaire pour les méthodes à petit et à grand pas.

De plus, on a présenté quelques résultats numériques pour montrer la validité de notre approche en faisant une comparaison avec d'autres méthodes basées sur différentes fonctions noyaux.

Certains sujets intéressants font l'objet de recherches actuelles :

- Tout d'abord, les directions de recherche utilisées sont toutes basées sur le schéma de symétrisation Nesterov Todd. Il est possible de concevoir des algorithmes similaires

- utilisant d'autres schémas de symétrisation et d'obtenir la complexité polynômiale liée.
- Numériquement : La détermination de la valeur de p dans le choix dynamique du pas α de façon à avoir la meilleure complexité.
 - Etendre ces recherches aux problèmes de complémentarité linéaire (LCP) et complémentarité semi-définie, la programmation quadratique et la programmation convexe en général.
 - Trouver de nouvelles fonctions noyaux et insister sur leur structure analytique afin de réduire le nombre des itérations internes et donc améliorer la complexité algorithmique.

Bibliographie

- [1] M. Achache. *A new primal-dual path-following method for convex quadratic programming*. Computational and Applied Mathematics. Vol 25(1) : 97-110, (2006).
- [2] M. Achache. *Complexity analysis of an interior point algorithm for the semidefinite optimization based on a kernel function with a double barrier term*. Acta Mathematica Sinica. English Series, 31 (3) : 543-556, (2015).
- [3] 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).
- [4] M. Achache & L. Guerra. *A full Nesterov-Todd-step feasible primal-dual interior point algorithm for convex quadratic semi-definite optimization*. Applied Mathematics and Computation. 231 : 581-590 (2014).
- [5] M. Achache & N. Tabchouche. *A full Nesterov-Todd step primal-dual path-following interior-point algorithm for semidefinite linear complementarity problems*. Croatian Operational Research Review. 9(1) : 37-50 (2018).
- [6] M. Achache & N. Tabchouche. *Complexity analysis and numerical implementation of large-update interior-point methods for SDLCP based on a new parametric barrier kernel function*. Optimization. 67 : 1-20 (2018).
- [7] F. Alizadeh. *Interior point methods in semidefinite programming with applications to combinatorial optimization*. SIAM journal on Optimization 5 : 13-51, (1995).

- [8] F. Alizadeh, J. P. A. Haeberly & M. L. Overton. *Primal-dual interior-point methods for semidefinite programming : Convergence rates, stability, and numerical results*. Technical Report. Computer Science Department, Courant Institute of Mathematical Sciences, New York University, (1996).
- [9] N. Anane. Méthodes de points intérieurs pour la programmation linéaire basées sur les fonctions noyaux. Thèse de Magister, Université Ferhat Abbas, Sétif1, (2012).
- [10] Y. Q. Bai, M. El Ghami & C. Roos. *A new efficient large-update primal-dual interior-point method based on finite barrier*. SIAM Journal on optimization. 13 (3) :766-782, (2003).
- [11] Y. Q. Bai & C. Roos. *A primal-dual interior point method based on a new kernel function with linear growth rate*. Proceedings of the 9th Australian Optimization Day, Perth, Australia, (07-11-2002).
- [12] Y. Q. Bai, M. El Ghami & C. Roos. *A comparative study of kernel function for primal-dual interior point algorithms in linear optimization*. SLAM journal on optimization. 15 (1) : 101-128, (2004).
- [13] Y. Q. Bai, G. Q. Wang & C. Roos. *Primal-dual interior point interior point algorithms for second-order cone optimization based on kernel functions*. Nonlinear Analysis.70 : 3548-3602, (2009).
- [14] A. Ben-Tal & A . Nemirovski. *Lectures on modern convex optimization. Analysis, Algorithms, Engineering Applications*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, (2001).
- [15] D. Benterki. Résolution des problèmes de programmation semi-définie par des méthodes de reduction du potentiel. Thèse de Doctorat, Département de Mathématiques, Université Ferhat Abbas, Sétif, Algérie, (2004).
- [16] M. Bierlaire. Introduction à l'optimisation différentiable, Press polytechniques et universitaires romandes, (2006).

- [17] M. Bouafia, D. Benterki & A. Yassine. *Complexity analysis of interior point methods for linear programming based on a parameterized kernel function*. RAIRO-Operations Research 50 (4-5), 935-949, (2016).
- [18] M. Bouafia, D. Benterki & A. Yassine. *An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a trigonometric barrier term*. J. of Optimization Theory and Applications, 170, No 2 : 528-545, (2016).
- [19] M. Bouafia, D. Benterki & A. Yassine. *An efficient parameterized logarithmic kernel function for linear optimization*. Optimization Letters, 12 (5), 1079-1097, (2018).
- [20] M. Bouafia & A. Yassine. *An efficient twice parameterized trigonometric kernel function for linear optimization*. Optimization and Engineering, pp.1-22, (2019).
- [21] S. Boyd & L. Vandenberghe. *Semidefinite programming*. SIAM Review, vol. 38(1) : pp. 49-96, (1996).
- [22] B. K. Choi & G. M. Lee. *On complexity analysis of the primal-dual interior-point method for semidefinite optimization problem based on a new proximity function*. Nonlinear Analysis. 71 (12), (2009).
- [23] 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).
- [24] I. Dikin. *Iterative solution of problems of linear and quadratic programming*. Doklady Akademii Nauk SSSR, vol. 174 :pp. 747-748, 1967. Translated in : Soviet Mathematics Doklady, 8 : 674-675, (1967).
- [25] I. Dikin. *On the convergence of an iterative process*. Upravlyaemye Sistemi. (In Russian), vol. 12 : pp. 54-60, (1974).
- [26] I. Dikin. *Letter to the editor*. Mathematical Programming, vol. 41 : pp. 393-394, (1988).

- [27] 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*. Journal of Computational and applied Mathematics, 236, 3613–3623, (2012).
- [28] M. El Ghami, G. Wang & T. Steihaug. *Primal-dual algorithms for semidefinite optimization problems based on kernel function with trigonometric barrier term*. International Journal of Applied Mathematics, Volume 32 No. 2 : 333-356, (2019).
- [29] M. El Ghami. New primal-dual interior-point methods based on kernel functions. Phd Thesis. Delft University, Netherland, (2005).
- [30] A.V. Fiacco, G.P. McCormick. Nonlinear programming : sequential unconstrained minimization techniques. John Wiley and Sons, New York, (1968).
- [31] R. Frisch. The logarithmic potential method for solving linear programming problems. Memorandum, University Institute of Economics, Oslo, (1955).
- [32] R. Frisch. *La résolution des problèmes de programmation linéaire par la méthode du potentiel logarithmique*. Cahiers du séminaire d'économie, vol. 4 : pp. 7-20 (1956).
- [33] F. Glineur. Etude des méthodes de point intérieur appliquées à la programmation linéaire et à la programmation semidéfinie, Document https://perso.uclouvain.be/francois.glineur/oldwww/courses/inma_2471/Reference_IPM.pdf
- [34] R. A. Horn & R. J. Charles. Matrix analysis, Compridge University Press, UK, (1986).
- [35] P. Huard. *Resolution of mathematical programming with nonlinear constraints by the method of centers*. In J. Abadie, editor, Nonlinear Programming, North Holland, Amsterdam, The Netherlands : pp. 207-219, (1967).
- [36] N. Karmarkar. *A new polynomial-time algorithm for linear programming*. Combinatorica, vol. 4 : pp. 373-395, (1984).
- [37] Z. Kebbiche. Etude et extentions d'algorithmes de points intérieurs pour la programmation non linéaire, Thèse de Doctorat, Université Farhat Abbas - Sétif, (2007).

- [38] A. Keraghel. Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar, Thèse de Doctorat, Université de Joseph Fourier, Grenoble I, France, (1989).
- [39] A. Keraghel. Analyse convexe. Théorie fondamentale et exercices, Editions Dar el'Houda, Ain M'lila, Algérie, (1999).
- [40] L. G. Khachiyan. *A polynomial algorithm in linear programming*. Doklady Akademiia Nauk SSSR, vol. 244 : pp. 1093-1096, (1979).
- [41] L. Liu & S. Li. *A new kind of simple kernel function yielding good iteration bounds for primal-dual interior-point methods*. Journal of Computational and Applied Mathematics, (2011).
- [42] R. D. C. Monteiro. *Polynomial convergence of primal-dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions*. SIAM Journal on Optimization, 8 : 797-812, (1998).
- [43] Y. Nesterov & M. Todd. *Self-scaled barriers and interior-point methods for convex programming*. Mathematics of Operations Research, vol. 22 (1) : pp. 1-42, (1997).
- [44] Y. Nesterov & A. S. Nemirovskii. Interior point polynomial algorithms in convex programming : Theory and Applications, SIAM, Philadelphia, PA, (1994).
- [45] J. Peng, C. Roos & T. Terlaky. *Self-regular functions and new search directions for linear and semidefinite optimization*. Mathematical Programming, vol. 93 : 129-171, (2002).
- [46] J. Peng, C. Roos & T. Terlaky. Self-Regularity. A new paradigm for primal-dual interior-point algorithms, Princeton University Press, (2002).
- [47] J. Peng, C. Roos & T. Terlaky. *A new class of polynomial primal-dual methods for linear and semidefinite optimization*. European Journal of operations research. 143 (2) : 234-256, (2002).

- [48] M. R. Peyghami & S. Fathi Hafshejani. *Complexity analysis of an interior point algorithm for linear optimization based on a new proximity function*. Numerical Algorithms, 67 : 33–48, (2014).
- [49] M. R. Peyghami, S. Fathi Hafshejani & L. Shirvani. *Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function*. Journal of Computational and Applied Mathematics, 255 : 74–85, (2014).
- [50] Z.G.Qian & Y.Q.Bay. *Primal-dual interior-point algorithms with dynamic step size based on kernel functions for linear programming*. Journal of Shanghai University, 9 , no. 5 : 391–396, (2005).
- [51] C. Roos, T.Terlaky & J-P.Vial. Theory and algorithms for linear optimization : an interior point approach, John Wiley & Son Ltd, (1997).
- [52] M.J. Todd. *Recent developments and new directions in linear programming*. In M. Iri and K. Tanabe, Eds. Mathematical Programming : Recent Developments and Applications. Kluwer Academic Press, Dordrecht : 109-157 (1989).
- [53] M.J. Todd, K.C.Toh & R.H.Tütüncü. *On the Nesterov-Todd direction in semidefinite programming*. Mathematical programming. Copyright(c) by the Society for Industrial and Applied Mathematics. SIAM Journal on Optimization, 8 : 769-796 (1998).
- [54] G. Q. Wang , Y. Q. Bai & C. Roos. *Primal-dual interior point algorithms for semidefinite optimization based on a simple kernel function*. Journal of Mathematical Modelling and Algorithms, 4 (4) : 409–433, (2005).
- [55] H. Wolkowicz, R. Saigal & L. Vandenberghe. " Handbook of semidefinite programming : Theory, algorithms, and applications [M]". Dordrecht : Kluwer Academic Publishers, (2000).
- [56] S. J. Wright. Primal-dual interior-point methods. SIAM Publications, Philadelphia, (1997).

- [57] M. W. ZHANG. *A large-update interior-point algorithm for convex quadratic semidefinite optimization based on a new kernel function*. Acta Mathematica Sinica, English Series, Vol. 28, No. 11 : pp. 2313–2328, (2012).
- [58] Y. ZHANG. *On extending primal-dual interior-point algorithms from linear programming to semidefinite programming*. Department of Mathematics and Statistics, University of Maryland, Baltimore, Technical Report TR : 20-95, (1995).

ملخص:

في هذه الأطروحة، نقدم طريقة النقاط الداخلية من نوع المسار المركزي لحل مسائل البرمجة الخطية والبرمجة النصف معرفة. لهذه الدراسة، نقترح فئة جديدة لدوال النواة التي تملك حد حاجز مزدوج. نعطي تكلفة خوارزميات ذات خطوة كبيرة وصغيرة. هذه الدراسة متنوعة باختبارات عددية لإظهار فعالية هذه الخوارزميات.

الكلمات المفتاحية :

طرق النقاط الداخلية؛ البرمجة الخطية؛ البرمجة النصف معرفة؛ خوارزمية ذات خطوة كبيرة وصغيرة؛ دالة النواة؛ تكلفة خوارزمي.

Abstract :

In this thesis, we present an interior point method of type central trajectory to solve Problems of linear and semidefinite programming. For this study, we propose a new class of kernel functions that have a double barrier term. We give the complexity for large and small update algorithms. This study is followed by numerical tests to show the efficiency of these algorithms.

Keywords :

Interior point methods; linear programming; Semidefinite programming; Large and small update algorithm; Kernel function; Algorithmic complexity.

Résumé :

Dans cette thèse, on présente une méthode de points intérieurs de type trajectoire centrale pour résoudre les problèmes de la programmation linéaire et la programmation semi-définie. Pour cette étude, on propose une nouvelle classe de fonctions noyaux qui possèdent un terme barrière double. On donne la complexité des algorithmes à grand et à petit pas. Cette étude est suivie par des tests numériques pour montrer l'efficacité de ces algorithmes.

Mots clés :

Méthodes de points intérieurs; Programmation linéaire; Programmation semi-définie; Algorithme à grand et à petit pas; Fonction noyau; Complexité algorithmique.