

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



THESE

Présentée par :

KETTAB Samia

Pour obtenir le titre de Doctorat en Sciences

OPTION

MATHÉMATIQUES APPLIQUÉES

THEME

**Généralisation d'une méthode de trajectoire centrale de points intérieurs pour
la programmation semi-définie**

Soutenue Le :14/10/2015

Devant le jury composé de :

Président :	Mr. M. ACHACHE	Prof. U.F.A. Sétif-1
Rapporteur :	Mr. D. BENTERKI	Prof. U.F.A. Sétif-1
Examineurs :	Mr. R. BENZINE	Prof. U.B.M. Annaba
	Mr. S. BOUROUBI	Prof. U.S.T.H.B. Alger

Année 2014/ 2015

Remerciements

Mes premiers remerciements vont à Monsieur **Dj. BENTERKI**, Professeur à l'Université Sétif-1 pour avoir encadré cette thèse et pour ses conseils pertinents. Je tiens à souligner ses qualités scientifiques, qui m'ont permis de mener à bien cette expérience.

Je remercie Monsieur **M. ACHACHE**, Professeur à l'Université Sétif-1, de m'avoir fait l'honneur de faire partie de ce jury et d'en être le président.

Je remercie aussi Monsieur **R. BENZINE** Professeur à l'Université d'Annaba et Monsieur **S. BOUROUBI** Professeur à l'Université de Bab Ezzouar pour avoir accepté de participer au jury de cette thèse.

Je remercie également, Monsieur **B. MERIKHI** Professeur à l'Université Sétif-1, pour ses encouragements.

Mes remerciements vont à mes proches et l'ensemble des amis et à toute l'équipe administrative du département de Mathématiques à l'Université Sétif-1.

Et enfin, Merci à mon mari pour son soutien, sa compréhension et son sacrifice durant l'élaboration de cette thèse.

Table des matières

Introduction	4
1 Analyse convexe et programmation mathématique	8
1.1 Matrices	8
1.1.1 Produit scalaire et normes	9
1.1.2 Matrices (semi-) définies positives	9
1.2 Analyse convexe	11
1.2.1 Ensembles affines	11
1.2.2 Ensembles convexes	12
1.2.3 Cônes convexes	14
1.2.4 Fonctions convexes	15
1.3 Programmation mathématique	18
1.3.1 Définitions	18
1.3.2 Classification	19
1.3.3 Principaux résultats d'existence et d'unicité	20
1.3.4 Conditions d'optimalité	21
1.4 Programmation linéaire	21
1.4.1 Méthodes de résolution d'un programme linéaire	23
2 Programmation semi-définie	34
2.1 Position du problème	34

2.1.1	Problème primal	34
2.1.2	Problème dual	35
2.2	Domaines d'applications	38
2.2.1	Problèmes de min-max des valeurs propres	38
2.2.2	Norme spectrale d'une matrice	39
2.2.3	Programmation quadratique avec des contraintes quadratiques	39
2.2.4	Problème de programmation non linéaire	41
2.3	Dualité en programmation semi-définie	42
2.3.1	Dualité faible	42
2.3.2	Dualité forte	46
2.4	Complémentarité en SDP	46
2.5	Méthodes de résolution	47
2.5.1	Méthodes de réduction du potentiel	47
2.5.2	Méthodes de trajectoire centrale de type primal-dual	49
3	Méthode de trajectoire centrale pour la programmation semi-définie	52
3.1	Introduction	52
3.2	Pénalisation logarithmique	53
3.2.1	Etude du problème perturbé $(SDP)_\mu$	54
3.2.2	Conditions d'optimalité pour $(SDP)_\mu$	56
3.3	Méthode de trajectoire centrale	57
3.3.1	Principe de la méthode de trajectoire centrale	57
3.3.2	Algorithme de trajectoire centrale T_μ	59
3.4	Relaxation du paramètre μ	60
3.4.1	Calcul de la direction	62
3.4.2	Calcul du pas de déplacement	65
3.4.3	Algorithme de trajectoire centrale T_W	67
3.4.4	Convergence de l'algorithme	68
3.5	Tests Numériques	70

3.5.1	Exemples à taille fixe	70
3.5.2	Exemples à taille variable	74
3.5.3	Commentaires	75
	Conclusion	76
	Bibliographie	77

Introduction

Le problème de la programmation semi-définie (SDP) est un domaine d'actualité dans la programmation mathématique, la plupart des documents sur SDP ont été écrits dans les années 90, bien que ses racines remontent à quelques décennies de plus, (voir par exemple Bellman et Fan [7]). Un article sur la programmation semi-définie paru en 1981 intitulé "Linear Programming with Matrix Variables" (Craven et Mond [14]), représente un ouvrage le plus convenable et la meilleure façon d'introduire le problème SDP. L'objectif est de minimiser le produit scalaire de deux matrices symétriques, à savoir une matrice constante C et une matrice variable X semi-définie positive ($X \succeq 0$), soumis à un ensemble de contraintes :

$$(SDP) \begin{cases} \min [\langle C, X \rangle = tr(CX)] \\ tr(A_i X) = b_i, \quad i = 1, \dots, m \\ X \succeq 0 \end{cases}$$

où tr , désigne la trace d'une matrice carrée, $A_i, b_i, i = 1, \dots, m$ sont respectivement des matrices symétriques réelles et des scalaires réels.

Le problème SDP est une généralisation de la programmation linéaire PL, où les vecteurs sont remplacés par des matrices et l'orthant positif (\mathbb{R}_+^n), est remplacé par le cône des matrices semi-définies positives. Ce qui explique donc le transport du savoir faire de la programmation linéaire à la programmation semi-définie.

Dans la dernière décennie, SDP a été l'un des domaines de recherche les plus actifs dans la programmation mathématique. Il y a deux principaux facteurs qui sont res-

ponsables de cet accru intérêt pour SDP. Premièrement, SDP intervient dans différents problèmes pratiques et mathématiques de grande importance, à savoir : la théorie du contrôle, l'optimisation combinatoire, la programmation non linéaire, le problème de coupe maximale dans la théorie des graphes et le problème de min-max des valeurs propres. Deuxièmement, la renaissance des méthodes de points intérieurs après les investigations effectuées par Karmarkar [24] pour la programmation linéaire PL.

Le domaine des méthodes de points intérieurs pour PL a commencé avec l'algorithme d'ellipsoïde de Khachiyan en 1979, qui a permis de lier le nombre d'itérations par un polynôme. Ce qui a répondu à la question de savoir si des problèmes de programmation linéaire peuvent être résolus en temps polynomial, mais les expériences pratiques par la méthode d'ellipsoïde ont été décevants.

Suivi par le célèbre article de Karmarkar [24] en 1984, qui a introduit un algorithme avec une complexité polynomiale améliorée, ceci a également été accompagnée par l'efficacité du comportement numérique de l'algorithme. Dans la décennie suivante, des milliers de documents sont apparus sur ce sujet.

Les méthodes de points intérieurs ont fait récemment l'objet de plusieurs monographies dont Roos, Terlaky et Vial [39], Wright [45], Ye [46] et Nesterov et Nemirovskii [35]. Un excellent survol des méthodes de points intérieurs pour la programmation linéaire est celui de Gonzaga [19]. Il a fallu presque dix ans pour étayer les prétentions de l'efficacité des méthodes de points intérieurs ; plusieurs études ont indiqué que ces méthodes ont des performances supérieures aux algorithmes de type simpliciale, l'état de l'art sur les problèmes de grande taille (voir par exemple Lustig et al. [30] et plus récemment E. D. Andersen et K. D. Andersen [4]).

Ayant des liens entre PL et SDP, on n'est pas surpris que les algorithmes de points intérieurs pour PL ont été étendus avec succès à SDP. La première extension des algorithmes de points intérieurs de PL à SDP a été réalisée par Nesterov et Nemirovski [35], et indépendamment par Alizadeh [1] en 1991. Cela explique le regain d'intérêt de la recherche dans SDP dans les années 1990. Dans ses travaux [1, 2], Alizadeh a étendu la

méthode projective de réduction du potentiel de Ye à partir de PL à SDP. D'autre part, Nesterov et Nemirovski [35] ont présenté une théorie profonde et unifiée des méthodes de points intérieurs pour résoudre les problèmes d'optimisation coniques plus généraux en utilisant la notion des fonctions barrières auto-concordantes.

Récemment, Kojima, Shindoh et Hara [28], Monteiro [32] et Y. Zhang [47] ont présenté des algorithmes primaux-duaux de points intérieurs pour la programmation semi-définie (ou pour la version de complémentarité linéaire semi-définie issue du problème SDP) qui sont généralisés à partir des algorithmes similaires conçus pour la programmation linéaire PL (ou pour le problème de complémentarité linéaire (PCL)). Leurs directions de recherche sont obtenues à partir d'une équation modifiée (en utilisant un facteur de paramétrisation) de Newton pour approcher une solution réalisable sur le chemin central. La convergence polynômiale de ces algorithmes a été étudiée et prouvée par plusieurs chercheurs [20, 21, 22, 31, 34, 36].

Dans notre travail, nous sommes intéressés à une extension d'une méthode de type trajectoire centrale en programmation linéaire à la programmation semi-définie [32]. L'idée principale des méthodes de trajectoire centrale est de remplacer le problème initial SDP par une suite de problèmes $(\text{SDP})_\mu$, paramétrisés par un facteur $\mu > 0$ via une fonction barrière logarithmique. La limite de la suite de solution optimale de $(\text{SDP})_\mu$ lorsque $\mu \rightarrow 0$, est une solution optimale du problème SDP. Dans cette thèse, nous relaxons le problème paramétrisé $(\text{SDP})_\mu$ par un autre problème paramétrisé $(\text{SDP})_W$ en utilisant un nouveau paramètre représenté par le rayon spectral d'une matrice diagonale semi-définie positive W , afin de donner plus de souplesse à l'étude théorique et numérique du problème paramétrisé obtenu et d'accélérer la convergence de l'algorithme correspondant.

La thèse est divisée en trois chapitres organisés comme suit :

Dans le premier chapitre, on présente une introduction de certaines notions et résultats qui seront utiles dans la suite, à savoir : l'analyse matricielle, l'analyse convexe et la programmation mathématique. Le chapitre 2, est un état de l'art sur la programmation semi-définie, à savoir : sa théorie, ses applications, sa dualité et les algorithmes de

résolution.

Dans le troisième chapitre, nous étudions et montrons l'existence et l'unicité de la solution optimale du problème paramétrisé $(\text{SDP})_\mu$ et $(\text{SDP})_W$. Nous donnons par la suite la description de l'algorithme correspondant à $(\text{SDP})_\mu$. Puis nous présentons l'algorithme obtenu en relaxant le paramètre μ par le rayon spectrale d'une matrice diagonale W correspondant au problème $(\text{SDP})_W$. A la fin de ce chapitre, nous présentons des tests numériques d'ordre comparatif, afin de porter plus de jugement sur les propos théoriques. Enfin, ce manuscrit est achevé par une conclusion et une bibliographie.

Chapitre 1

Analyse convexe et programmation mathématique

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.

1.1 Matrices

On notera $M_{m,n}(\mathbb{R})$ l'espace vectoriel des matrices réelles $m \times n$ et $M_n(\mathbb{R})$ l'espace vectoriel des matrices carrées d'ordre n , (i.e., $M_n(\mathbb{R}) = M_n(\mathbb{R})$). L'espace vectoriel des matrices carrées symétriques d'ordre n est noté S_n .

Définition 1.1.1 *La trace d'une matrice $A \in M_n(\mathbb{R})$ est la somme de ses éléments diagonaux,*

$$\text{tr}(A) = \sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i(A),$$

où λ_i sont les valeurs propres de A .

On a la propriété importante suivante :

$$\forall A, B \in M_n(\mathbb{R}), \operatorname{tr}(AB) = \operatorname{tr}(BA).$$

1.1.1 Produit scalaire et normes

On définit tout d'abord un produit scalaire sur l'ensemble $M_n(\mathbb{R})$ des matrices de taille $n \geq 1$, à coefficients dans \mathbb{R} :

Si $A, B \in M_n(\mathbb{R})$, le produit scalaire de A et B , noté $\langle A, B \rangle$ est défini par :

$$\langle A, B \rangle = \operatorname{tr}(B^T A) = \sum_{i,j=1}^n a_{ij} b_{ij} = \operatorname{tr}(A^T B).$$

A ce produit scalaire, on associe une norme, dite de Frobenius :

$$\|A\|_F = \|A\| = \sqrt{\langle A, A \rangle} = \sqrt{\operatorname{tr}(A^T A)} \text{ pour tout } A \in M_n(\mathbb{R}).$$

Pour $A, B \in S_n$, on a :

$$\langle A, B \rangle = \operatorname{tr}(AB) \text{ et } \|A\| = \sqrt{\sum_{i=1}^n \lambda_i^2(A)},$$

où les λ_i sont les valeurs propres de A .

Proposition 1.1.1 *Toutes les valeurs propres d'une matrice symétrique $A \in S_n$ sont réelle et de plus il existe une matrice orthogonale $P \in M_n(\mathbb{R})$ qui diagonalise A , (i.e., $P^T A P = D_A$ où D_A est une matrice diagonale dont les éléments diagonaux sont les valeurs propres de A).*

1.1.2 Matrices (semi-) définies positives

Définition 1.1.2 *Soit $A \in S_n$*

– A est dite semi-définie positive si

$$\forall x \in \mathbb{R}^n, \quad x^T A x \geq 0,$$

et l'on notera $A \in S_n^+$ ou $A \succeq 0$.

– A est dite définie positive si

$$\forall x \in \mathbb{R}^n, \quad x \neq 0, \quad x^T A x > 0,$$

et l'on notera $A \in S_n^{++}$ ou $A \succ 0$.

Toute sous-matrice principale d'une matrice symétrique semi-définie (resp. définie) positive est aussi semi-définie (resp. définie) positive.

En particulier, tous les éléments diagonaux d'une matrice symétrique semi-définie (resp. définie) positive sont positifs (resp. strictement positifs).

Lemme 1.1.1 Soient $A, B \in S_n^+$. Alors $\langle A, B \rangle \geq 0$ et en plus $\langle A, B \rangle = 0$ si et seulement si $AB = 0$.

Proposition 1.1.2 Soit $B \in M_n(\mathbb{R})$ une matrice régulière. Alors

$$A \in S_n^+ \Leftrightarrow B^T A B \in S_n^+,$$

et

$$A \in S_n^{++} \Leftrightarrow B^T A B \in S_n^{++}.$$

Théorème 1.1.1 Pour $A \in S_n$, les propriétés suivantes sont équivalentes :

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

On a aussi, pour une matrice définie positive, l'existence d'une racine carrée :

Proposition 1.1.3 *Soit $A \in S_n^{++}$. Alors il existe une matrice unique $B \in S_n^{++}$ telle que*

$$A = B^2.$$

On a de plus : $\text{rg}(A) = \text{rg}(B)$ et on notera $B = A^{\frac{1}{2}}$.

Théorème 1.1.2 *(Complément de Schur)*

Soient $A \in S_m^{++}$, $C \in S_n$ et $B \in M_{m,n}$. On a alors les équivalences suivantes :

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \Leftrightarrow C - B^T A^{-1} B \succeq 0,$$

et

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succ 0 \Leftrightarrow C - B^T A^{-1} B \succ 0.$$

1.2 Analyse convexe

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

1.2.1 Ensembles affines

Définition 1.2.1 *Un sous ensemble F de \mathbb{R}^n est dit affine si :*

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

On dit aussi que F est une variété affine (ou linéaire).

Les ensembles affines élémentaires sont : \emptyset , $\{x\}$ ($x \in \mathbb{R}^n$), et chaque sous-espace vectoriel de \mathbb{R}^n .

Définition 1.2.2 On appelle combinaison affine des éléments x_1, \dots, x_m de \mathbb{R}^n tout élément de la forme :

$$x = \sum_{i=1}^m \lambda_i x_i, \text{ avec } \lambda_i \in \mathbb{R} \text{ et } \sum_{i=1}^m \lambda_i = 1.$$

Théorème 1.2.1 Toute partie affine de \mathbb{R}^n contient ses combinaisons affines :

$$\forall x_i \in F, \sum_{i=1}^m \lambda_i x_i \in F, \text{ avec } \lambda_i \in \mathbb{R} \text{ et } \sum_{i=1}^m \lambda_i = 1.$$

Définition 1.2.3 Soit S une partie de \mathbb{R}^n . Alors il existe une partie affine unique $F \subseteq \mathbb{R}^n$ contenant S appelée enveloppe affine de S et notée $\text{aff}(S)$, c'est la plus petite partie affine de \mathbb{R}^n contenant S .

$$\text{aff}(S) = \cap \{F_S : S \subset F_S \text{ et } F_S \text{ affine}\}.$$

Si $S \neq \emptyset \Rightarrow \text{aff}(S) \neq \emptyset$ (puisque $S \subseteq \text{aff}(S)$).

Théorème 1.2.2 Soit $S \subseteq \mathbb{R}^n$, $S \neq \emptyset$, alors :

$$\text{aff}(S) = \left\{ x \in \mathbb{R}^n / x = \sum_{i=1}^m \lambda_i x_i, x_i \in S, \sum_{i=1}^m \lambda_i = 1 \right\}$$

1.2.2 Ensembles convexes

Définition 1.2.4 Un sous ensemble C de \mathbb{R}^n est dit convexe si :

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

Autrement dit, si le segment de droite joignant deux points quelconques x, y est en-

tièrement inclus dans C ,

$$[x, y] = \{\lambda x + (1 - \lambda)y \mid 0 \leq \lambda \leq 1\} \subset C.$$

Les opérations algébriques suivantes conservent la convexité

- L'intersection quelconque.
- Le produit cartésien.
- Les transformations affines.
- Les combinaisons linéaires $\sum_{i=1}^m \alpha_i C_i$, ($\alpha_i \in \mathbb{R}$ et $m \in \mathbb{N}$).
- La translation $C + a$ / $a \in \mathbb{R}^n$.

Définition 1.2.5 On appelle combinaison convexe de m -vecteurs de \mathbb{R}^n , x_1, \dots, x_m , toute combinaison linéaire :

$$\sum_{i=1}^m \lambda_i x_i, \text{ où } \lambda_i \geq 0 \text{ et } \sum_{i=1}^m \lambda_i = 1.$$

Définition 1.2.6 L'enveloppe convexe de l'ensemble $S \subset \mathbb{R}^n$ est le plus petit convexe de \mathbb{R}^n contenant S .

$$\text{conv}(S) = \cap(C_i),$$

avec C_i convexe contenant S .

Et on a :

$$S \text{ convexe} \Leftrightarrow S = \text{conv}(S).$$

Définition 1.2.7 Soit C un convexe non vide de \mathbb{R}^n et $x \in C$.

Le point x est dit extrémal (ou sommet de C) s'il n'est pas à l'intérieur d'un segment de droite contenu dans C . Autrement dit si :

$$\forall x_1, x_2 \in C, \forall \lambda \in]0, 1[, x = \lambda x_1 + (1 - \lambda)x_2 \Rightarrow x = x_1 = x_2.$$

Remarque 1.2.1 La définition d'un point extrémal x d'un convexe C , est équivalente à chacune des deux propriétés suivantes :

1. $x = \frac{1}{2}x_1 + \frac{1}{2}x_2 \Rightarrow x = x_1 = x_2 \quad (\forall x_1, x_2 \in C)$.
2. $C - \{x\} = \{y \in C \mid y \neq x\}$ est convexe.

Remarque 1.2.2 *Tout point extrémal d'un convexe C est un point de la frontière de C .*

Définition 1.2.8 (Intérieur relatif)

L'intérieur relatif d'un sous ensemble non vide C de \mathbb{R}^n , noté $ri(C)$ est défini comme l'intérieur de C vu comme sous-ensemble de $aff(C)$.

$$ri(C) = \{x \in aff(C) \mid \exists \varepsilon > 0 : B(x, \varepsilon) \cap aff(C) \subseteq C\}.$$

C est dit relativement ouvert si $ri(C) = C$.

1.2.3 Cônes convexes

Définition 1.2.9 *Un ensemble $K \subset \mathbb{R}^n$ est un cône si*

$$\mathbb{R}_+^* K \subseteq K,$$

ou

$$\forall x \in K, \forall \lambda > 0, \lambda x \in K.$$

Un cône est dit pointé si $K \cap (-K) = \{0\}$, où $-K = \{-x : x \in K\}$.

Si K est convexe, on l'appelle cône convexe.

Exemple 1.2.1 S_n^+ est un cône convexe pointé dans S_n .

Proposition 1.2.1

- *L'intersection quelconque de cônes convexes est un cône convexe.*
- *Le produit cartésien de cônes convexes est un cône convexe.*

Cônes de récession

Soit C un convexe non vide de \mathbb{R}^n et $a \in C$, on pose

$$C_\infty(a) = \{d \in \mathbb{R}^n : a + \lambda d \in C, \forall \lambda > 0\}.$$

Alors, $C_\infty(a)$ est un cône convexe non vide.

Définition 1.2.10 On appelle cône de récession (ou asymptôte) de C l'ensemble :

$$C_\infty = \bigcap_{a \in C} C_\infty(a).$$

Un élément $d \in C_\infty$ est appelé direction de récession.

Proposition 1.2.2 Soit C un convexe fermé non vide de \mathbb{R}^n , alors :

$$C \text{ est borné ssi } C_\infty = \{0\}.$$

1.2.4 Fonctions convexes

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, on associe à f les ensembles suivants :

- **Epigraphe de f :**

$$epi(f) = \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} / f(x) \leq \alpha\}.$$

- **Domaine effectif de f :**

$$dom(f) = \{x \in \mathbb{R}^n / f(x) < +\infty\}.$$

- **Ensembles de niveau $\alpha \in \mathbb{R}$:**

$$S_\alpha(f) = \begin{cases} \{x \in \mathbb{R}^n / f(x) \leq \alpha\} & (\text{inférieur large}), \\ \{x \in \mathbb{R}^n / f(x) \geq \alpha\} & (\text{supérieur large}). \end{cases}$$

Définition 1.2.11 f est dite propre si

$$\text{dom} f \neq \emptyset \text{ et } f(x) > -\infty, \forall x \in \mathbb{R}^n.$$

Dans le cas contraire, on dit que f est impropre.

Définition 1.2.12 On dit que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ définie sur un ensemble convexe $C \subset \mathbb{R}^n$ est convexe si $\forall x, y \in C, \forall \lambda \in [0, 1]$ on a :

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

ou d'une manière équivalente : si $\forall m \in \mathbb{N}^*, \forall \lambda_i \geq 0, \sum_{i=1}^m \lambda_i = 1, x_i \in C$

$$f\left(\sum_{i=1}^m \lambda_i x_i\right) \leq \sum_{i=1}^m \lambda_i f(x_i).$$

f est strictement convexe sur C si

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

$$\forall x, y \in C, x \neq y, \forall \lambda \in]0, 1[.$$

Définition 1.2.13 Une fonction est dite concave si $(-f)$ est convexe.

Théorème 1.2.3 Si f est deux fois continument différentiable sur un ouvert Ω et C une partie convexe de Ω , les conditions suivantes sont équivalentes :

1. f est convexe sur C .
2. $\forall x, y \in C, f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$, où $\nabla f(x)$ est le vecteur gradient de f au point x .
3. $\forall x \in C$, la matrice Hessienne $H(x) = \nabla^2 f(x)$ est semi-définie positive dans C .

Lemme 1.2.1 Une combinaison linéaire à coefficients positifs des fonctions convexes est une fonction convexe.

$$\left. \begin{array}{l} f_1, \dots, f_m \text{ convexe} \\ \lambda_1 \geq 0, \dots, \lambda_m \geq 0 \end{array} \right\} \Rightarrow f = \sum_{i=1}^m \lambda_i f_i \text{ convexe.}$$

Lemme 1.2.2 Soit $(f_i)_{i \in I}$ une famille quelconque de fonctions convexes. Alors leur enveloppe supérieure :

$$f(x) = \sup_{i \in I} f_i(x),$$

est une fonction convexe.

Définition 1.2.14 Soit C fermé non vide de \mathbb{R}^n et $f : C \rightarrow]-\infty, +\infty]$.

f est dite inf-compacte si ses ensembles de niveau inférieurs $S_\lambda(f)$ sont compacts $\forall \lambda \in \mathbb{R}$.

Lemme 1.2.3 f est inf-compacte si et seulement si $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$.

Définition 1.2.15 On appelle fonction asymptote (ou de recession) la fonction convexe f_∞ dont l'épigraphe est $(\text{epi}(f))_\infty$.

Lemme 1.2.4

- Pour $a \in \text{dom}(f)$, $d \in \mathbb{R}^n$

$$f_\infty(d) = \sup_{t>0} \frac{f(a+td) - f(a)}{t} = \lim_{t \rightarrow +\infty} \frac{f(a+td) - f(a)}{t}.$$

- $S_0(f_\infty) = [S_\lambda(f)]_\infty$, $\forall \lambda$ tel que $S_\lambda(f) \neq \{0\}$.

Théorème 1.2.4 [26]

$$f \text{ est inf-compacte} \Leftrightarrow S_0(f_\infty) = \{0\}.$$

Remarque 1.2.3 Il est d'usage d'étendre une fonction f définie sur un ensemble C de \mathbb{R}^n en une fonction $g : \mathbb{R}^n \rightarrow]-\infty, +\infty]$ comme suit :

$$g(x) = \begin{cases} f(x) & \text{si } x \in C \\ +\infty & \text{sinon.} \end{cases}$$

Si C est convexe alors :

f est convexe (strictement convexe) sur C si et seulement si son extension g est convexe (strictement convexe) sur \mathbb{R}^n .

1.3 Programmation mathématique

1.3.1 Définitions

- **Problème d'optimisation**

On définit un problème d'optimisation avec contraintes comme suit :

$$(P) \begin{cases} \min f(x) \\ x \in C, \end{cases}$$

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ continue, $C \subseteq \mathbb{R}^n$ est l'ensemble des contraintes.

Si $C = \mathbb{R}^n$, (P) est appelé problème d'optimisation sans contraintes.

- **Programme mathématique**

En général, un programme mathématique est défini comme suit :

$$(PM) \begin{cases} \min f(x) \\ x \in D \subseteq \mathbb{R}^n, \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continue, $D \subseteq \mathbb{R}^n$ l'ensemble des contraintes présentées souvent comme :

$$D = \{x \in \mathbb{R}^n / f_i(x) \leq 0, i = 1, \dots, m, g_j(x) = 0, j = 1, \dots, p\},$$

avec $f_i, g_j : \mathbb{R}^n \rightarrow \mathbb{R}$.

- **Solution réalisable (faisable) :** On appelle solution réalisable de (PM) tout point x^0 vérifiant les contraintes (i.e., $x^0 \in D$).
- **Solution optimale globale :** On appelle solution optimale globale toute solution réalisable (notée x^*) qui minimise f sur D .

L'ensemble des solutions optimales globales est noté :

$$\arg \min_D f(x).$$

- **Solution optimale locale :** Un point $x^* \in D$ est une solution optimale locale de (PM) s'il existe un voisinage ϑ de x^* tel que :

$$f(x^*) \leq f(x); \forall x \in \vartheta.$$

L'ensemble des solutions optimales locales de (PM) est noté :

$$\text{locmin}_D f(x).$$

1.3.2 Classification

On classe le problème (PM) à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité des fonctions du problème.

À ce propos,

- (PM) est convexe si les fonctions f, f_i sont convexes et les fonctions g_j sont affines.
- (PM) est différentiable si les fonctions f, f_i, g_j sont différentiables.

Qualification des contraintes

Définition 1.3.1 On dit que la contrainte $f_i(x) \leq 0$ est active ou saturée en $\bar{x} \in D$ si $f_i(\bar{x}) = 0$.

On introduit alors l'ensemble

$$I(\bar{x}) = \{i : f_i(\bar{x}) = 0\}.$$

Par définition, une contrainte d'égalité est saturée.

Voici trois conditions de qualification classiques :

- Karlin (1959) : Si D est un polyèdre convexe (i.e., f_i, g_j sont affines), alors les contraintes sont qualifiées en tout point réalisable.
- Slater (1950) : Si D est convexe (i.e., f_i convexe et g_j affine) et $\text{int}(D) \neq \phi$, alors les contraintes sont qualifiées en tout point réalisable.
- Mangasarian-Fromovitz (1967) : Si les gradients de toutes les fonctions des contraintes saturées en $\bar{x} \in D$ sont linéairement indépendants, alors les contraintes sont qualifiées en \bar{x} .

Théorème 1.3.1 *Pour un programme convexe, tout optimum local est un optimum global.*

1.3.3 Principaux résultats d'existence et d'unicité

Théorème 1.3.2 (Weierstrass)

Si D est compact (fermé et borné) dans \mathbb{R}^n et f est continue sur D , alors (PM) admet au moins une solution optimale globale $x^ \in D$.*

Corollaire 1.3.1 *Si D est non vide et fermé, et f est continue et coercive sur D (au sens que $f(x) \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$), alors (PM) admet une solution optimale globale.*

Corollaire 1.3.2 *Si f est une fonction inf-compacte, alors (PM) admet une solution optimale globale.*

Théorème 1.3.3 *Si D est convexe et f est strictement convexe, alors il existe au plus une solution optimale de (PM).*

1.3.4 Conditions d'optimalité

Conditions nécessaires du premier ordre : multiplicateurs de Karush-Kuhn-Tucker

Théorème 1.3.4 *Supposons qu'une des conditions précédentes de qualification des contraintes est satisfaite au point $\bar{x} \in D$. Une condition nécessaire pour que f admet un minimum local en \bar{x} est qu'il existe $\lambda_i \in \mathbb{R}_+^m$ et $\mu \in \mathbb{R}^p$ tels que :*

$$\begin{cases} \nabla f(\bar{x}) + \sum_{i=1}^m \lambda_i \nabla f_i(\bar{x}) + \sum_{j=1}^p \mu_j \nabla g_j(\bar{x}) = 0 \\ \lambda_i f_i(\bar{x}) = 0, \quad i = 1, \dots, m \\ g_j(\bar{x}) = 0, \quad j = 1, \dots, p. \end{cases}$$

Les λ_i et les μ_j sont appelés multiplicateurs de Karush-Kuhn-Tucker.

Remarque 1.3.1

- Si les contraintes ne sont pas qualifiées en \bar{x} , les conditions de KKT ne s'appliquent pas.
- Si (PM) est convexe, les conditions de KKT sont à la fois nécessaires et suffisantes pour que \bar{x} soit un minimum global.

1.4 Programmation linéaire

Un problème de programmation linéaire consiste à minimiser une fonction linéaire sous contraintes linéaires. On considère les programmes linéaires sous la forme standard :

$$(PL) \begin{cases} \min c^T x \\ Ax = b \\ x \geq 0, \end{cases}$$

où A est une matrice réelle ($m \times n$) de plein rang ($rgA = m < n$),

$b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

Le dual de PL est donné par :

$$(DL) \begin{cases} \max b^T y \\ A^T y \leq c \\ y \in \mathbb{R}^m. \end{cases}$$

On note par :

- $F_P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, l'ensemble des solutions primales réalisables de PL qui est un polyèdre convexe fermé.
- $F_P^\circ = \{x \in \mathbb{R}^n : Ax = b, x > 0\}$, l'ensemble des solutions primales strictement réalisables de PL .
- $F_D = \{(y, z) \in \mathbb{R}^{m \times n} : A^T y + z = c, z \geq 0\}$, l'ensemble des solutions duales réalisables de DL .
- $F_D^\circ = \{(y, z) \in \mathbb{R}^{m \times n} : A^T y + z = c, z > 0\}$, l'ensemble des solutions duales strictement réalisables de DL .
- $F_P \times F_D$, l'ensemble des solutions primales-duales réalisables de PL et DL .
- $F_P^\circ \times F_D^\circ$, l'ensemble des solutions primales-duales strictement réalisables de PL et DL .

Théorème 1.4.1 (Dualité faible)

Soient $(x, y) \in F_P \times F_D$ deux solutions réalisables de PL et DL , respectivement. Alors,

$$b^T y \leq c^T x.$$

La quantité $c^T x - b^T y \geq 0$ s'appelle *saut de dualité*.

Théorème 1.4.2 (Dualité forte)

Soient $(x, y) \in F_P \times F_D$ deux solutions réalisables de PL et DL respectivement tel que :

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

Alors x et y sont des solutions optimales de PL et DL .

Corollaire 1.4.1

- Si $c^T x$ est non borné inférieurement sur F_P alors DL est non réalisable (l'ensemble des contraintes duales F_D est vide).
- Si $b^T y$ est non borné supérieurement alors PL est non réalisable (i.e., F_P est vide).
- Si PL est non réalisable alors DL est non borné (et réciproquement).

Proposition 1.4.1 *Un programme linéaire réalisable et borné (objectif borné) possède au moins une solution optimale, situé sur la frontière du domaine réalisable.*

Définition 1.4.1

- 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 $x = (x_B, x_N) \in \mathbb{R}^n$ 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 F_P si et seulement s'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).

Proposition 1.4.2 *Si un programme linéaire possède une solution optimale, alors au moins un sommet du domaine réalisable est une solution optimale.*

1.4.1 Méthodes de résolution d'un programme linéaire

Méthode de simplexe

Elle est développée à la fin des années 40 par G. Dantzig. Elle tient compte systématiquement des résultats établis précédemment. Elle évolue sur la frontière du domaine

réalisable de sommet en sommet adjacents, en réduisant la valeur de l'objectif jusqu'à

l'optimum. Un critère d'optimalité simple permet de reconnaître le sommet optimal. Le nombre de sommets étant fini, l'algorithme ainsi défini converge en un nombre fini d'itération n'excédant pas le nombre $C_n^m = \frac{n!}{m!(n-m)!}$; sous l'hypothèse que tous les sommets visités sont non dégénérés.

Dans le cas dégénéré l'algorithme risque de cycler, cependant il existe des techniques convenables pour éviter ce phénomène. En général, la méthode du simplexe possède un comportement numérique très satisfaisant confirmé par ses applications multiples dans la résolution d'une large classe de problèmes pratiques.

En théorie; la méthode n'a pas autant de succès, elle est plutôt jugée inefficace de par sa complexité arithmétique exponentielle de l'ordre de $O(2^n)$ opérations.

Méthode de points intérieurs

On désigne par méthode de points intérieurs, toute procédure de résolution générant une suite de points appartenant à l'intérieur relatif du domaine réalisable et convergeant vers une solution optimale.

Ces méthodes sont réputées grâce à leur convergence polynomiale, leur rapidité et efficacité et se sont révélées comme de véritables concurrentes des méthodes simpliciales surtout pour les problèmes de grandes tailles.

On distingue trois classes fondamentales de méthodes de points intérieurs à savoir : les méthodes affines, les méthodes projectives de réduction du potentiel et les méthodes de trajectoire centrale.

1) Méthode affine

C'est une méthode itérative (dû à dikin 1967) qui évolue dans l'intérieur relatif du domaine réalisable. En effet, à partir d'un point x^0 strictement réalisable i.e.,

$$x^0 > 0 / Ax^0 = b,$$

on construit une suite $\{x^k\}$ qui converge vers l'optimum sous certaines conditions. Le passage d'un itéré à l'autre se fait moyennant un déplacement convenable suivant une direction de descente calculée à partir d'un problème normalisé qui centre le point courant qu'on transforme ensuite dans l'espace original. L'algorithme correspondant à cette méthode est d'une structure simple, malheureusement la polynomialité est très difficile à établir.

2) Méthodes projectives avec potentiel

Elles sont typiquement construites pour trouver des solutions améliorantes du problème non linéaire suivant :

$$\left\{ \begin{array}{l} \min_{x,y,z} \left[f(x,y,z) = q \log(c^T x - b^T y) - \sum_{j=1}^n \log(x_j) \right] \\ Ax = b, x \geq 0 \\ A^T y + z = c, z \geq 0 \end{array} \right.$$

La fonction objectif $f(x, y, z)$ est appelée fonction potentiel primale-duale et q son paramètre. Ces méthodes visent à ramener la valeur de $f(x, y, z)$ à $-\infty$. C'est ce type de fonction que Karmarkar introduit dans son article [24], la fonction potentiel est une intermédiaire pour réduire progressivement le saut de dualité à zéro. Ces méthodes ne sont pas aussi simples que les méthodes affines mais on peut établir la polynomialité de l'algorithme correspondant.

3) Méthodes primale-duale de type trajectoire centrale (TC)

Elles sont introduites à la même époque que les méthodes projectives avec potentiel (au début des années 90), elles possèdent des propriétés théoriques très intéressantes : complexité polynomiale, convergence asymptotique superlinéaire et éventuellement un statut numérique prometteur.

La trajectoire centrale du programme linéaire PL est définie à partir de sa version

barrière :

$$(PL)_\mu \begin{cases} \min \left[c^T x - \mu \sum_{j=1}^n \log(x_j) \right] \\ Ax = b \\ x > 0, \end{cases}$$

où μ est un paramètre barrière. Le problème $(PL)_\mu$ étant convexe et différentiable, à contraintes linéaire (donc qualifiées partout), les conditions nécessaires et suffisantes d'optimalité de (KKT) s'écrivent :

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

qui est équivalent au système :

$$\begin{cases} A^T y + z - c = 0, z > 0 \\ Ax = b, x > 0 \\ XZe - \mu e = 0, \end{cases} \quad (1.1)$$

où $X = \text{diag}(x_0, \dots, x_n)$, $X^{-1} = \text{diag}\left(\frac{1}{x_0}, \dots, \frac{1}{x_n}\right)$, $z = \mu X^{-1} e > 0$ et $Z = \text{diag}(z_0, \dots, z_n)$

Désignant par $(x(\mu), y(\mu), z(\mu))$ la solution du système non linéaire (1.1).

Pour $\mu > 0$ donné, l'ensemble :

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

est appelé la trajectoire centrale de $(PL)_\mu$.

La stratégie des méthodes de trajectoire centrale, consiste à chercher des solutions approchées du système non linéaire (1.1), suivant la trajectoire centrale.

Saut de dualité :

Le saut de dualité est donné par :

$$c^T x - b^T y = n\mu.$$

Facteur de centralité :

Pour mesurer la qualité de la solution trouvée, on introduit un facteur dit de «centralité» ou de proximité défini par le scalaire suivant :

$$\delta(x, z, \mu) = \|XZe - \mu e\|.$$

A ce propos, un point est dit voisin de la trajectoire centrale s'il appartient à l'ensemble suivant :

$$S_{cent}(\sigma) = \{(x, y, z) \in F_P^\circ \times F_D^\circ / \delta(x, z, \mu) \leq \sigma\mu, 0 < \sigma < 1\}.$$

Le système (1.1) est un système d'équations non linéaires. Il est résolu par la méthode de Newton appliquée à la fonction $F(x, y, z) = 0$

où $F : \mathbb{R}^{2n+m} \longrightarrow \mathbb{R}^{2n+m}$ définie par :

$$F(x, y, z) = \begin{pmatrix} A^T y + z - c \\ Ax - b \\ XZe - \bar{\mu}e \end{pmatrix},$$

pour chaque $\bar{\mu} = \sigma\mu$ ($0 < \sigma < 1$) et $(x, y, z) \in F_P^\circ \times F_D^\circ$.

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

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

La matrice jacobienne de F au point (x, y, z) est définie par :

$$\nabla F(x, y, z) = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix}.$$

Par un simple calcul, on arrive à résoudre le système linéaire suivant :

$$\begin{cases} A^T \Delta y + \Delta z = 0 \\ A \Delta x = 0 \\ X \Delta z + Z \Delta x = XZ e - \bar{\mu}, \end{cases} \quad (1.2)$$

qui donne :

$$\begin{cases} \Delta x = \left[Z^{-1} - Z^{-1} X A^T (A Z^{-1} X A^T)^{-1} A Z^{-1} \right] (Xz - \bar{\mu} e) \\ \Delta y = - \left[(A Z^{-1} X A^T)^{-1} A Z^{-1} \right] (Xz - \bar{\mu} e) \\ \Delta z = -A^T \Delta y. \end{cases}$$

Le nouveau itéré s'écrit alors sous la forme :

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

On réitère jusqu'à l'obtention d'une valeur $\bar{\mu}$ suffisamment proche de zéro.

Remarque 1.4.1 *Théoriquement, on suppose que la solution initiale est strictement réalisable et proche de la trajectoire centrale, mais l'obtention de ce point est une difficulté majeure. Pour remédier, Ding[16] et Kebbiche[25] ont introduit le paramètre poids associé à la fonction barrière.*

4) Méthode de trajectoire centrale avec poids

On considère le problème linéaire sous forme standard suivant :

$$(PL) \begin{cases} \min c^T x \\ Ax = b \\ x \geq 0. \end{cases}$$

On associe à PL le problème perturbé défini comme suit :

$$(PL)_{\mu r} \begin{cases} \min \left[c^T x - \mu \sum_{j=1}^n r_j \log(x_j) \right] \\ Ax = b \\ x > 0, \end{cases}$$

$\mu > 0$, $r = (r_1, \dots, r_n)^T \in \mathbb{R}_+^n$ est le vecteur de poids associé à la fonction barrière. Les conditions nécessaires et suffisantes d'optimalité de (KKT) s'écrivent :

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

Ou encore sous la forme :

$$\begin{cases} A^T y + z - c = 0 \\ Ax = b, x > 0 \\ Xz - \mu r = 0, z > 0, \end{cases} \quad (1.3)$$

où l'on a posé : $z = \mu X^{-1} r$.

En appliquant de nouveau la méthode de Newton, on obtient le système :

$$\begin{cases} A^T \Delta y + \Delta z = 0 \\ A \Delta x = 0 \\ X \Delta z + Z \Delta x = -Xz + \mu r. \end{cases} \quad (1.4)$$

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

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

Le point (x^+, y^+, z^+) est voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$S_{cent}(\sigma r) = \{(x, y, z) \in F_P^\circ \times F_D^\circ / \|XZe - \mu r\| \leq \sigma \mu, 0 < \sigma < 1\}.$$

Algorithme

TCSP : désigne la méthode de trajectoire centrale (classique) sans poids.

TCAP : désigne la méthode de trajectoire centrale avec poids.

Début

$\epsilon > 0$ est un paramètre de précision. et $(x, y, z) \in F_P^\circ \times F_D^\circ$,

on pose $\sigma = \frac{\|X^0 Z^0 e\|}{\sqrt{n}}$ et $r = \frac{X^0 Z^0 e}{\sigma}$.

Soient $\eta = \min \{r_i\}$, $R = \text{diag}(r_i)$ pour $i = 1, \dots, n$, $\theta = \left(1 - \frac{\sqrt{2}}{2}\right) \eta$, $\beta = 1 - \frac{\theta}{\sqrt{n}}$

et $\mu^0 = \frac{x^{0t} z^0}{n}$.

Si $\left\| \frac{X^0(\mu)^n s^0(\mu)}{\mu} - e \right\| \leq \theta$ alors : Aller à **TCSP**.

Sinon Aller à **TCAP**.

Fin si

TCSP :

$k = 0$

Tant que $(\mu^k > \epsilon)$ faire :

1. Prendre $\mu^{k+1} = \beta \mu^k = \beta \frac{x^{k^t} z^k}{n}$
2. Calculer la direction $(\Delta x^k, \Delta y^k, \Delta z^k)$; en résolvant le système (1.3).
3. Calculer : $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + (\Delta x^k, \Delta y^k, \Delta z^k)$.
4. Prendre $k = k + 1$.

Fin tant que

TCAP :

$k = 0$,

Tant que $(\mu^k > \epsilon)$ faire :

1. Prendre $\mu^{k+1} = \beta \mu^k = \beta \frac{x^{k^T} R z^k}{n}$.
2. Calculer la direction $(\Delta x^k, \Delta y^k, \Delta z^k)$; en résolvant le système (1.3).
3. Calculer : $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + (\Delta x^k, \Delta y^k, \Delta z^k)$.
4. Prendre $k = k + 1$.

Fin tant que

Fin.

Resultat de convergence de la méthode de trajectoire centrale avec poids

Théorème 1.4.3 [25] Soit $\theta = \sigma = \left(1 - \frac{\sqrt{2}}{2}\right)\eta$, supposons que $(x, y, z) \in S_{cent}(\sigma r)$ et

$$\mu^+ = \beta\mu,$$

$$\text{tel que } \beta = 1 - \frac{\theta}{\sqrt{n}}, \mu = \frac{x^T Rz}{n} \text{ alors :}$$

- 1) $(x^+, y^+, z^+) \in S_{cent}(\sigma r)$.
- 2) $\frac{x^{+T} Rz^+}{n} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) \frac{x^T Rz}{n}$.

Théorème 1.4.4 [15] Pour $\epsilon > 0$, l'algorithme converge après $K = O\left(\frac{2\sqrt{n}}{(2 - \sqrt{2})\eta} |\log \epsilon|\right)$ itérations. avec $\eta = \min\{r_i\}$.

Récemment, plusieurs variantes de type trajectoire centrale primale-duale basées sur la notion des fonctions noyaux sont proposées par des chercheurs [5, 6, 11, 17, 37, 38], afin d'améliorer le comportement de l'algorithme de trajectoire centrale classique. L'introduction de la fonction noyau dans le système (1.2) permet de trouver de nouvelles directions qui offrent une complexité algorithmique d'ordre $O\left(\sqrt{n} \log n \log \frac{n}{\epsilon}\right)$ pour les algorithmes à petit pas et d'ordre $O\left(n \log \frac{n}{\epsilon}\right)$ pour ceux à grand pas.

Nous définissons le vecteur v pour tout vecteur réalisable primal $x > 0$, et le vecteur réalisable dual $z > 0$,

$$v = \sqrt{\frac{xz}{\mu}}; \quad xz = (x_1 z_1, \dots, x_n z_n)^T. \quad (1.5)$$

On définit les vecteurs normés de la direction de déplacement d_x et d_z comme suit :

$$d_x = \frac{v \Delta x}{x} \text{ et } d_z = \frac{v \Delta z}{z}. \quad (1.6)$$

Utilisant (1.5) et (1.6) on peut récrire le système (1.2) comme suit :

$$\begin{cases} \bar{A}d_x = 0 \\ \bar{A}^t d_y + d_z = 0 \\ d_x + d_z = v^{-1} - v, \end{cases} \quad (1.7)$$

où $\bar{A} = \frac{1}{\mu}AV^{-1}X$, $V = \text{diag}(v)$.

Notons que le coté droit de la troisième équation de (1.7) est le gradient de la fonction barrière logarithmique $\Phi_l(v)$,

$$d_x + d_z = -\nabla\Phi_l(v), \quad (1.8)$$

où

$$\Phi_l(v) = \sum_{i=1}^n \psi_l(v_i).$$

On appelle ψ_l la fonction noyau de la fonction logarithmique barrière $\Phi_l(v)$.

Les résultats prometteurs obtenus par cette méthode ont encouragé les chercheurs d'étendre la notion des fonctions noyaux à d'autres types de problèmes d'optimisation tel que les problèmes de programmation semi-définie.

Chapitre 2

Programmation semi-définie

Dans ce chapitre, nous aborderons l'étude des problèmes de minimisation de fonctions linéaires à contraintes linéaires, où la variable est une matrice carrée symétrique $X \in S_n$ semi-définie positive.

2.1 Position du problème

2.1.1 Problème primal

Le problème primal de la programmation semi-définie sous forme standard s'écrit :

$$(SDP) \begin{cases} \min \langle C, X \rangle \\ AX = b \\ X \in S_n^+, \end{cases}$$

où : $b \in \mathbb{R}^m$ et A est un opérateur linéaire de S_n dans \mathbb{R}^m tel que

$$AX = \begin{pmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{pmatrix},$$

C et $A_i, i = 1, \dots, m$ sont des matrices dans $M_n(\mathbb{R})$. Sans perte de généralité on peut supposer que C et les A_i sont symétriques.

Définition 2.1.1 Une matrice $X \in S_n$ est dite réalisable (strictement réalisable resp) si :

$$AX = b, X \in S_n^+ \quad (AX = b, X \in S_n^{++} \text{ resp}),$$

et on note par

$$\mathcal{F}_P = \{X \in S_n^+ : AX = b\},$$

l'ensembles des solutions réalisables de SDP.

Et

$$\mathcal{F}_P^\circ = \{X \in S_n^{++} : AX = b\},$$

l'ensembles des solutions strictement réalisables de SDP.

Définition 2.1.2 La valeur optimale primale de SDP est définie par :

$$val(P) = \inf [\langle C, X \rangle : AX = b, X \in S_n^+].$$

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

$$X^* \in \mathcal{F}_P \text{ et } \langle C, X^* \rangle = val(P).$$

2.1.2 Problème dual

On introduit l'opérateur $A^* : \mathbb{R}^m \rightarrow S_n$ adjoint de A , qui est défini par

$$\forall X \in S_n, \forall y \in \mathbb{R}^m : \langle AX, y \rangle = \langle X, A^*y \rangle.$$

Ainsi

$$\langle AX, y \rangle = \sum_{i=1}^m y_i \text{tr}(A_i X) = \text{tr} \left(X \sum_{i=1}^m y_i A_i \right) = \langle X, A^* y \rangle,$$

donc

$$A^* y = \sum_{i=1}^m y_i A_i.$$

La méthode la plus simple pour obtenir le dual de SDP est d'utiliser la dualisation Lagrangienne de sa contrainte d'égalité. On définit la fonction lagrangienne associée à SDP :

$$L(X, y) = \langle C, X \rangle + \langle b - AX, y \rangle, \quad X \in S_n^+, \quad y \in \mathbb{R}^m,$$

et on considère la fonction duale,

$$\begin{aligned} D(y) &= \min_{X \in S_n^+} L(X, y) \\ &= \min_{X \in S_n^+} [\langle C, X \rangle + \langle b - AX, y \rangle] \\ &= \min_{X \in S_n^+} \left[\langle C, X \rangle + \sum_{i=1}^m (b_i - \langle A_i, X \rangle) y_i \right] \\ &= \min_{X \in S_n^+} \left[\left\langle C - \sum_{i=1}^m y_i A_i, X \right\rangle + \sum_{i=1}^m b_i y_i \right] \\ &= \min_{X \in S_n^+} \left[\left\langle C - \sum_{i=1}^m y_i A_i, X \right\rangle \right] + \sum_{i=1}^m b_i y_i \\ &= b^T y + \min_{X \in S_n^+} [\langle C - A^* y, X \rangle]. \end{aligned}$$

Il est facile de voir que

$$\min_{X \in S_n^+} [\langle C - A^* y, X \rangle] = \begin{cases} 0 & \text{si } C - A^* y \succeq 0 \\ -\infty & \text{sinon.} \end{cases}$$

D'où

$$\max_{y \in \mathbb{R}^m} D(y) = \begin{cases} \max_{y \in \mathbb{R}^m} b^T y & \text{si } C - A^* y \succeq 0 \\ -\infty & \text{sinon.} \end{cases}$$

Donc, le problème dual de SDP est :

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

Définition 2.1.3

- (y, Z) est une solution réalisable de SDD si :

$$(y, Z) \in \mathbb{R}^m \times S_n^+ : Z = C - \sum_{i=1}^m y_i A_i.$$

- (y, Z) est une solution strictement réalisable de SDD si :

$$(y, Z) \in \mathbb{R}^m \times S_n^{++} : Z = C - \sum_{i=1}^m y_i A_i.$$

On note par :

$$\begin{aligned} \mathcal{F}_D &= \left\{ (y, Z) \in \mathbb{R}^m \times S_n : Z = C - \sum_{i=1}^m y_i A_i, Z \succeq 0 \right\} \\ \mathcal{F}_D^\circ &= \left\{ (y, Z) \in \mathbb{R}^m \times S_n : Z = C - \sum_{i=1}^m y_i A_i, Z \succ 0 \right\}, \end{aligned}$$

Les ensembles de solutions réalisables et strictement réalisables de SDD , respectivement.

Définition 2.1.4 La valeur optimale duale de *SDD* est définie par :

$$\text{val}(D) = \sup \left[b^T y : C - \sum_{i=1}^m y_i A_i \in S_n^+, y \in \mathbb{R}^m \right].$$

(y^*, Z^*) est une solution optimale duale de *SDD* si :

$$(y^*, Z^*) \in \mathcal{F}_D \text{ et } b^T y^* = \text{val}(D).$$

Et on notera enfin :

$$\mathcal{F} = \mathcal{F}_P \times \mathcal{F}_D \text{ et } \mathcal{F}^\circ = \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ,$$

les ensembles de solutions réalisables primales-duales et strictement réalisables primales-duales de *SDP* et *SDD* respectivement.

2.2 Domaines d'applications

2.2.1 Problèmes de min-max des valeurs propres

Le problème suivant a été étudié dans [3, 21] comme suit :

$$\lambda^* = \min_{y \in \mathbb{R}^n} \lambda_{\max}(C + A(y)),$$

où $C \in S_n$ et

$$\begin{aligned} A : \mathbb{R}^n &\longrightarrow S_n \\ y &\longrightarrow A(y), \end{aligned}$$

est un opérateur linéaire.

Ce problème peut s'écrire sous la forme :

$$m_p = \min [\lambda : \lambda I - C - A(y) \in S_n, y \in \mathbb{R}^n, \lambda \in \mathbb{R}].$$

Son dual est le problème semi-défini (*SDP*) suivant :

$$m_d = \max [\langle C, X \rangle : X \in S_n, A^t(X) = 0, \text{tr}(X) = 1].$$

2.2.2 Norme spectrale d'une matrice

Soit $A \in M_n(\mathbb{R})$. On considère sa norme spectrale

$$\max_{\lambda \in SP(A^T A)} \sqrt{\lambda} = \|A\|_2$$

Cette norme peut être calculer à l'aide du problème (*SDP*) suivant :

$$\left\{ \begin{array}{l} \|A\|_2 = \min_{t \geq 0} t \\ \left[\begin{array}{cc} tI & A \\ A^T & tI \end{array} \right] \succeq 0. \end{array} \right.$$

On utilise ici le théorème du complément de Schur.

2.2.3 Programmation quadratique avec des contraintes quadratiques

La contrainte quadratique convexe

$$(Ax + b)^T (Ax + b) - c^T x - d \leq 0,$$

avec $x \in \mathbb{R}^k$, peut s'écrire comme :

$$\left[\begin{array}{cc} I & Ax + b \\ (Ax + b)^T & c^T x + d \end{array} \right] \succeq 0.$$

Cette dernière inégalité peut être exprimée par :

$$F(x) = F_0 + x_1 F_1 + \dots + x_k F_k \succeq 0,$$

avec

$$F_0 = \begin{bmatrix} I & b \\ b^T & d \end{bmatrix}, \quad F_i = \begin{bmatrix} 0 & a_i \\ a_i^T & c_i^T \end{bmatrix}, \quad i = 1, \dots, k,$$

où $A = [a_1, \dots, a_k]$. Par conséquent, un programme convexe quadratique avec des contraintes quadratiques de type :

$$(QCQP) \begin{cases} \min f_0(x) \\ f_i(x) \leq 0, \quad i = 1, \dots, L, \end{cases}$$

où chaque f_i est une fonction quadratique convexe :

$$f_i(x) = (A_i x + b)^T (A_i x + b) - c_i^T x - d_i,$$

peut être formulé comme suit :

$$(QCQP) \begin{cases} \min t \\ \begin{pmatrix} I & A_0 x + b \\ (A_0 x + b)^T & c_0^T x + d_0 + t \end{pmatrix} \succeq 0 \\ \begin{pmatrix} I & A_i x + b \\ (A_i x + b)^T & c_i^T x + d_i + t \end{pmatrix} \succeq 0, \quad i = 1, \dots, L. \end{cases}$$

$(QCQP)$ est un programme semi-défini avec les variables $x \in \mathbb{R}^k$ et $t \in \mathbb{R}$.

2.2.4 Problème de programmation non linéaire

Soit le programme non linéaire suivant :

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

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

$$\begin{cases} \min \langle C, X \rangle \\ X \in S_2 \\ \langle A, X \rangle = b, \end{cases}$$

$$\text{avec } C = I, X = \begin{pmatrix} x_1^3 & x_3 \\ x_3 & x_2 \end{pmatrix}, A = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}, b = 1.$$

Pour des besoins théoriques et numériques, on suppose dans tout ce qui suit :

Hypothèse 1 : Les matrices A_i ($i = 1, \dots, m$) sont linéairement indépendantes.

Sous l'hypothèse 1, y est uniquement déterminé pour Z donnée. Il s'agit essentiellement de la même hypothèse que dans *PL* où la matrice des contraintes doit être de plein rang.

Une autre hypothèse nécessaire dans toute la suite concerne la stricte faisabilité.

Hypothèse 2 : Les ensembles \mathcal{F}_P° et \mathcal{F}_D° sont non vides (i.e., $\exists X \in \mathcal{F}_P$ et $(y, Z) \in \mathcal{F}_D$ tel que $X \succ 0$ et $Z \succ 0$).

Notons que l'hypothèse 2 est compatible avec la définition usuelle de régularité de Slater pour l'optimisation convexe, si nous utilisons le fait que l'intérieur relatif est donné par les matrices symétriques définies positives.

2.3 Dualité en programmation semi-définie

Définition 2.3.1 (*Saut de dualité*) Soit $(X, y, Z) \in \mathcal{F}$, la quantité

$$\langle C, X \rangle - b^T y,$$

est appelée *saut de dualité des deux problèmes SDP et SDD en (X, y, Z)* .

2.3.1 Dualité faible

Théorème 2.3.1 Si $X \in \mathcal{F}_P$ et $(y, Z) \in \mathcal{F}_D$ alors :

$$\langle C, X \rangle - b^T y = \langle Z, X \rangle \geq 0.$$

Preuve :

$$\begin{aligned} \langle C, X \rangle - b^T y &= \langle C, X \rangle - \sum_{i=1}^m b_i y_i \\ &= \langle C, X \rangle - \sum_{i=1}^m y_i \langle A_i, X \rangle \\ &= \left\langle C - \sum_{i=1}^m y_i A_i, X \right\rangle \\ &= \langle Z, X \rangle. \end{aligned}$$

Puisque les matrices X et Z sont toutes les deux semi-définies positives et d'après le Lemme 1.1.1 $\langle Z, X \rangle \geq 0$.

Contrairement à la programmation linéaire, il n'est pas toujours vrai que l'optimalité de *SDP* et *SDD* implique que $\langle Z, X \rangle = 0$ comme le montrent les exemples suivants :

Exemple 2.3.1 *Considérons le problème semi-défini suivant :*

$$(SDP) \begin{cases} \min \langle C, X \rangle \\ \langle A_i, X \rangle = b_i, \quad i = 1, 2 \\ X \in S_3^+, \end{cases}$$

avec :

$$C = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, X = \begin{pmatrix} x_{11} & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix},$$

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

Ce problème s'écrit aussi sous la forme suivante :

$$\left\{ \begin{array}{l} \min(1 - x_{12}) \\ \begin{pmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 - x_{12} \end{pmatrix} \in S_3^+ \end{array} \right.$$

L'ensemble des solutions réalisables est donné par :

$$\{(x_{12}, x_{22}) \in \mathbb{R}^2 : x_{12} = 0, x_{22} \geq 0\}.$$

L'ensemble des solutions optimales est :

$$\left\{ \begin{array}{l} \begin{pmatrix} 0 & 0 & 0 \\ 0 & x_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} ; x_{22} \geq 0 \end{array} \right\}.$$

Il result que $val(P) = 1$.

Le problème dual est

$$(SDD) \left\{ \begin{array}{l} \max b^T y \\ C - \sum_{i=1}^2 y_i A_i \in S_3^+ \\ y \in \mathbb{R}^2, \end{array} \right.$$

qui correspond à la forme :

$$(SDD) \left\{ \begin{array}{l} \max 2y_2 \\ \begin{pmatrix} -y_1 & -y_2 & 0 \\ -y_2 & 0 & 0 \\ 0 & 0 & 1 - 2y_2 \end{pmatrix} \in S_3^+ \end{array} \right.$$

L'ensemble des solutions réalisables est :

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

L'ensemble des solutions optimales est :

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

Il result que $\text{val}(D) = 0$.

Les problèmes SDP et SDD sont tous deux réalisables, leurs ensembles des solutions optimales sont tous deux non vides mais le saut de dualité est non nul. On conclut que la réalisabilité des deux problèmes SDP et son dual SDD ne suffit pas pour avoir les mêmes valeurs optimales.

Exemple 2.3.2 Considérons le problème semi-défini suivant :

$$(SDP) \left\{ \begin{array}{l} \min \langle C, X \rangle \\ \langle A_i, X \rangle = b_i, \quad i = 1, 2 \\ X \in S_2^+, \end{array} \right.$$

avec :

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix},$$

$$A_1 = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}, b = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

Ce problème s'écrit aussi sous la forme suivante :

$$\begin{cases} \min 2x_{12} \\ \begin{pmatrix} 1 & x_{12} \\ x_{12} & 0 \end{pmatrix} \in S_2^+. \end{cases}$$

Donc l'unique solution optimale est : $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ et $\text{val}(P) = 0$.

Le problème duale est

$$(SDD) \begin{cases} \max b^T y \\ C - \sum_{i=1}^2 y_i A_i \in S_2^+ \\ y \in \mathbb{R}^2, \end{cases}$$

qui correspond à la forme :

$$(SDD) \begin{cases} \max(-y_1) \\ \begin{pmatrix} y_1 & 1 \\ 1 & y_2 \end{pmatrix} \in S_2^+. \end{cases}$$

L'ensemble des solutions réalisables est :

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

On trouve $\text{val}(D) = 0$, mais l'ensemble des solutions optimales est vide.

En conclusion, $\text{val}(P) = \text{val}(D) = 0$ (fini); SDP et SDD sont tous deux réalisables, mais SDD n'admet pas de solutions optimales.

En conséquences, l'existence d'une solution optimale pour l'un des deux problèmes SDP ou son dual SDD n'implique pas nécessairement l'optimalité de l'autre même si leurs valeurs optimales sont identiques et finis. Pour conserver le resultat de forte dualité pour SDP , on exige la condition de stricte réalisabilité de l'un des deux problèmes comme le montre le théorème suivant.

2.3.2 Dualité forte

Théorème 2.3.2 [35]

1. Si $val(P) > -\infty$ et $\mathcal{F}_P^\circ \neq \phi$, alors l'ensemble des solutions optimales de SDD est non vide et borné et on a :

$$val(P) = val(D).$$

2. Si $val(D) < +\infty$ et $\mathcal{F}_D^\circ \neq \phi$, alors l'ensemble des solutions optimales de SDP est non vide et borné et on a :

$$val(P) = val(D).$$

3. Si $\mathcal{F}^\circ \neq \phi$, alors les ensembles des solutions optimales de SDP et SDD sont non vides et bornés et on a :

$$val(P) = val(D).$$

2.4 Complémentarité en SDP

A l'image de ce qui se passe en programmation linéaire, on peut exprimer la condition pour X^* et y^* d'être des solutions optimales de SDP et SDD sous la forme d'une condition dite de complémentarité.

Théorème 2.4.1 Soient $X^* \in \mathcal{F}_P$ et $(y^*, Z^*) \in \mathcal{F}_D$ de saut de dualité

$$\langle C, X^* \rangle - b^T y^* = \langle Z^*, X^* \rangle.$$

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

Le problème de complémentarité s'écrit alors :

$$(Pc) \begin{cases} \langle A_i, X \rangle & = b_i, & i = 1, \dots, m, X \succeq 0 \\ C - \sum_{i=1}^m y_i A_i & = Z, & Z \succeq 0 \\ XZ & = 0. \end{cases} \quad (2.1)$$

Remarque 2.4.1 Comme SDP est un problème convexe, la solution du problème de complémentarité est un optimum global pour SDP .

2.5 Méthodes de résolution

La similarité de SDP avec la programmation linéaire PL , a motivé les chercheurs d'appliquer des techniques ayant fait leurs preuves pour les PL , en particulier les méthodes de trajectoire centrale de type primal-dual de points intérieurs.

La généralisation des méthodes de points intérieurs de PL à SDP remonte au début des années 1990. Les premiers algorithmes dans ce sens ont été introduites de façon indépendante par Alizadeh [2] et Nesterov et Nemirovskii [35]. Alizadeh [2] a étendu l'algorithme projectif de réduction du potentiel de Ye à partir de PL à SDP et affirme que de nombreux algorithmes de points intérieurs connus pour PL peuvent être transformés en des algorithmes pour résoudre SDP . Nesterov et Nemirovski [35] ont présenté la théorie profonde des méthodes de points intérieurs qui est basée sur des fonctions barrières auto-concordantes.

2.5.1 Méthodes de réduction du potentiel

En 1995, Alizadeh a proposé une méthode primale-duale de réduction du potentiel [2]. En 2003 et inspirant des travaux de Alizadeh, Benterki [9] a fait une étude algorithmique

et numérique sur des méthodes de réduction du potentiel pour résoudre des problèmes de programmation semi-définie.

Les ingrédients principaux utilisés dans ce type d'algorithme sont :

1. Une transformation projective T_k qui permet de ramener SDP à la forme réduite plus maniable (ressemble à la forme réduite de Karmarkar en PL).

Plaçons nous à l'étape k , où l'on dispose de l'itéré $X_k \in S_n^{++}$. La factorisation de Cholesky de X_k donne une matrice triangulaire inférieure L_k dont les éléments diagonaux sont strictement positifs tel que $L_k L_k^t = X_k$. Alors, la transformation projective est définie comme suit :

$$\begin{aligned} T_k & : S_n \longrightarrow S_n \times \mathbb{R}^r \\ X & \mapsto (\bar{X}, \bar{x}) \end{aligned}$$

$$T_k(X) = (\bar{X}, \bar{x}),$$

où

$$\bar{X} = \frac{(n+r)L_k^{-1}X L_k^{-t}}{r + \langle X_k^{-1}, X \rangle}, \quad \bar{x} = \frac{(n+r)}{r + \langle X_k^{-1}, X \rangle} e_r.$$

Ici $e_r = (1, 1, \dots, 1)^T \in \mathbb{R}^r$. Notons que $X = T_k^{-1}(\bar{X}, \bar{x})$.

Le problème SDP sera équivalent au problème semi-défini suivant :

$$(TSDP) \left\{ \begin{array}{l} \min \langle \bar{C}, \bar{X} \rangle + \bar{c}^T(z) \bar{x} \\ \bar{\mathcal{A}} \begin{pmatrix} \text{vec} \bar{X} \\ \bar{x} \end{pmatrix} = \begin{pmatrix} 0 \\ n+r \end{pmatrix} \\ \|\bar{X} - I\|^2 + \|\bar{x} - e_r\|_2^2 \leq \beta^2, \end{array} \right.$$

où

- $\bar{C} = L_k^t C L_k \in S_n$.
- $\bar{c}(z) = -\begin{pmatrix} z \\ r \end{pmatrix} e_r$ est un vecteur de \mathbb{R}^r .
- $\bar{A}_i = L_k^t A_i L_k \in S_n$.

- $\bar{\mathcal{A}} = \mathcal{A}(L_k \otimes L_k)$ est une $m \times n^2$ -matrice.
- $\bar{A} = \begin{pmatrix} -1 \\ r \end{pmatrix} b e_r^t$ est une $m \times r$ -matrice.
- $\bar{\bar{\mathcal{A}}} = \begin{pmatrix} \bar{\mathcal{A}} & \bar{A} \\ (\text{vec}I)^t & e_r^t \end{pmatrix}$ est une $(m+1) \times (n^2+r)$ -matrice.

La solution optimale $(\bar{X}^*, \bar{x}^*)^t$ de $(TSDP)$ est facile à obtenir ; il suffit par exemple d'écrire les conditions d'optimalité, on trouve :

$$\begin{pmatrix} \text{vec} \bar{X}^* \\ \bar{x}^* \end{pmatrix} = \begin{pmatrix} \text{vec} I \\ e_r \end{pmatrix} - \beta \frac{p(z)}{\|p(z)\|},$$

où $p(z)$ est la projection du vecteur coût $\begin{pmatrix} \text{vec} \bar{C} \\ \bar{c}(z) \end{pmatrix}$ sur le noyau de la matrice $\bar{\bar{\mathcal{A}}}$.

2. Fonction potentielle primale-duale :

$$\psi(X, y) = (n+r) \log \langle X, C - \sum_{i=1}^m y_i A_i \rangle - \log \det(X(C - \sum_{i=1}^m y_i A_i)),$$

définie pour tout $X \in S_n^{++}$ et tout $y \in \mathbb{R}^m$ vérifiant $C - \sum_{i=1}^m y_i A_i \in S_n^{++}$.

Cette fonction permet de prouver la convergence polynomiale de l'algorithme proposé par Alizadeh [2].

2.5.2 Méthodes de trajectoire centrale de type primal-dual

En suivant la tendance en PL , les méthodes de trajectoires centrales primales-duales pour (SDP) ont suscité beaucoup d'attention. En 1996, Helemberg et al [21] ont proposé un algorithme primal-dual de trajectoire centrale pour (SDP) . En 1999, R. D. C. Monteiro and T. Tsuchiya ont proposé une méthode primale-duale de trajectoire centrale et ont montré la convergence polynomiale [31]. On trouve aussi les travaux de M. Halicka, E. De Klerk and C. Roos en 2002 sur la convergence de la méthode de trajectoire centrale pour SDP [20].

Le principe de ces méthodes est de minimiser le saut de dualité en résolvant le système suivant :

$$\begin{cases} C - \sum_{i=1}^m y_i A_i, & = Z \\ \langle A_i, X \rangle & = b_i, \quad i = 1, \dots, m \\ XZ & = \mu I, \quad X, Z \succ 0. \end{cases} \quad (2.2)$$

qui est le système paramétrisé des conditions d'optimalité des problèmes *SDP* et *SDD*. Le système (2.2) admet une solution unique sous les hypothèses 1 et 2, i.e., que les $A_i, i = 1, \dots, n$ sont linéairement indépendantes, et que $\mathcal{F}^\circ \neq \emptyset$. La solution du système est notée par $(X(\mu), y(\mu), Z(\mu))$ pour $\mu > 0$ fixé.

L'ensemble des solutions $(X(\mu), y(\mu), Z(\mu))$ pour tout $\mu > 0$ définit la trajectoire centrale qui converge vers la solution optimale quand μ tend vers 0.

Pour résoudre le système non linéaire (2.2), on utilise la méthode de Newton. L'objectif est d'obtenir des directions primales et duales $\Delta X, \Delta y$ et ΔZ , respectivement en résolvant le système linéaire :

$$\begin{cases} \langle A_i, \Delta X \rangle & = 0, & i = 1, \dots, m, \quad X \in S_n^{++} \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z & = 0, & Z \in S_n^{++} \\ X\Delta Z + Z\Delta X & = \mu I - XZ. \end{cases} \quad (2.3)$$

L'itération de Newton complète est définie par :

$$\begin{aligned} X^+ &= X + \Delta X, \\ y^+ &= y + \Delta y, \\ Z^+ &= Z + \Delta Z. \end{aligned}$$

et doit être strictement réalisable (i.e., $X^+ \in \mathcal{F}_P^\circ$ et $(y^+, Z^+) \in \mathcal{F}_D^\circ$).

Malheureusement, X^+ n'est pas toujours symétrique, pour remédier ce problème, plusieurs chercheurs ont proposé dans la littérature des directions symétriques. On cite

par exemples les travaux de Zhang [47] , Helemberg et al, Kojima et al et Monteiro [21, 28, 32], Alizadeh-Heaberly-Overton [3] et Nesterov-Todd [41].

Actuellement, plusieurs chercheurs [12, 18, 43, 44] ont étendu l'approche des fonctions noyaux de la programmation linéaire à la programmation semie-définie pour trouver de nouvelles directions et améliorer la complexité des algorithmes.

Dans le chapitre suivant, on présente et on montre d'une manière détaillée les résultats d'existence et d'unicité de la solution optimale du problème pèrturbé suivie par des modifications qu'on a apporté sur le paramètre μ afin de relaxer et améliorer le comportement de l'algorithme considéré.

Chapitre 3

Méthode de trajectoire centrale pour la programmation semi-définie

3.1 Introduction

Dans ce chapitre, nous nous intéressons à une extension de la méthode de trajectoire centrale primale-duale pour la programmation semi-définie SDP . Nous relaxons le problème perturbé du problème initial SDP par un nouveau paramètre afin de donner plus de souplesse à l'étude théorique et numérique du problème logarithmique barrière associé à SDP et d'accélérer la convergence de l'algorithme correspondant.

Rappelons que le problème de programmation semi-définie considéré est :

$$(SDP) \begin{cases} \min \langle C, X \rangle \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m \\ X \in S_+^n. \end{cases}$$

Auquel on associe son problème dual :

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

3.2 Pénalisation logarithmique

Définition 3.2.1 On appelle fonction barrière toute fonction f qui vérifie :

- f est à valeurs finies à l'intérieur relatif du domaine réalisable.
- $f \rightarrow +\infty$ quand x s'approche de la frontière.

Exemples :

$$1) f_1(X) = \begin{cases} \det X^{-\alpha} & \text{si } X \in S_{++}^n, \alpha > 0 \\ +\infty & \text{sinon.} \end{cases}$$

$$2) f_2(X) = \begin{cases} -\ln \det X & \text{si } X \in S_{++}^n \\ +\infty & \text{sinon.} \end{cases}$$

$$3) f_3(X) = \begin{cases} \text{tr}(X^{-1}) & \text{si } X \in S_{++}^n \\ +\infty & \text{sinon.} \end{cases}$$

Les fonctions barrières citées ci-dessus sont deux fois différentiables et on a :

$$1. \begin{cases} \nabla f_1(X) = -\alpha \det X^{-\alpha} X^{-1} \\ \nabla^2 f_1(X)H = \alpha \det X^{-\alpha} (\alpha \langle X^{-1}, H \rangle X^{-1} + X^{-1} H X^{-1}), \end{cases}$$

$$2. \begin{cases} \nabla f_2(X) = -X^{-1} \\ \nabla^2 f_2(X)H = X^{-1} H X^{-1}, \end{cases}$$

$$3. \begin{cases} \nabla f_3(x) = -X^{-2}, \\ \nabla^2 f_3(X)H = X^{-2} H X^{-1} + X^{-1} H X^{-2}. \end{cases}$$

Dans notre travail, on considère la fonction barrière définie sur le cône des matrices semi-définies S_n^+ par :

$$f_\mu(X) = \begin{cases} \langle C, X \rangle - \mu \ln \det(X) & \text{si } X \in S_n^{++}, \\ +\infty & \text{ailleurs.} \end{cases}$$

On associe à SDP , le problème perturbé $(SDP)_\mu$ suivant :

$$(SDP)_\mu \begin{cases} \min f_\mu(X) \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m \end{cases}$$

avec $\mu > 0$.

3.2.1 Etude du problème perturbé $(SDP)_\mu$

On commence par le lemme suivant.

Lemme 3.2.1 $\{Y \in S_n^+ : \langle C, Y \rangle \leq 0, \langle A_i, Y \rangle = 0, i = 1, \dots, m\} = \{0\}$.

Preuve :

Nous savons que l'ensemble des solutions optimales de SDP ($Sol(SDP)$) est convexe fermé borné et non vide. Par conséquent, son cône de récession $Sol(SDP)^\infty$ est réduit au singleton $\{0\}$. Mais

$$Sol(SDP)^\infty = \{Y \in S_n^+ : \langle C, Y \rangle \leq 0, \langle A_i, Y \rangle = 0, i = 1, \dots, m\} = \{0\}.$$

Théorème 3.2.1 *Le problème $(SDP)_\mu$ admet une solution optimale unique.*

Preuve :

1) Existence : *Il suffit de prouver que le cône de récession de l'ensemble convexe fermé*

$$S_\lambda = \{X : \langle A_i, X \rangle = b_i \quad \forall i, \quad f_\mu(X) \leq \lambda\} \quad \lambda \in \mathbb{R}.$$

est réduit à $\{0\}$. On a :

$$S_\lambda^\infty = \{Y : \langle A_i, Y \rangle = 0, i = 1, \dots, m\} \cap \{f_\mu^\infty(Y) \leq 0\}.$$

Calculons la fonction de récession f_μ^∞ . Soit $X \in S_n^{++}$, alors pour $Y \in S_n$,

$$f_\mu^\infty(Y) = \lim_{t \rightarrow \infty} \frac{f_\mu(X + tY) - f_\mu(X)}{t}.$$

Il est clair que si $Y \notin S_n^+$, pour t assez grand, $X + tY$ n'appartient pas à S_n^+ et donc $f_\mu(Y) = +\infty$. Supposons que $Y \in S_n^+$. Alors :

$$f_\mu^\infty(Y) = \langle C, Y \rangle - \mu \lim_{t \rightarrow \infty} \frac{\ln \det(X + tY) - \ln \det(X)}{t}.$$

Puisque X est définie positive alors il existe une matrice symétrique définie positive Z tel que $X = Z^2$. Il existe une matrice P et une matrice diagonale semi-définie D telles que

$$Z^{-1}YZ^{-1} = P^TDP \text{ avec } P^T P = I.$$

Il en résulte que

$$\det(X + tY) = \det(ZP^t(I + tD)PZ) = \det(X) \prod_{i=1}^n (I + td_i),$$

où les d_i sont les éléments diagonaux de la matrice D . Donc

$$f_\mu^\infty(Y) = \langle C, Y \rangle - \mu \sum_{i=1}^n \lim_{t \rightarrow \infty} \frac{\ln(1 + td_i)}{t} = \langle C, Y \rangle.$$

Comme

$$S_\lambda^\infty = \{Y : \langle A_i, Y \rangle = 0 \forall i\} \cap \{f_\mu^\infty(Y) \leq 0\},$$

il sensuit :

$$S_\lambda^\infty = \{Y \in S_+^n : \langle C, Y \rangle \leq 0, \langle A_i, Y \rangle = 0, i = 1, \dots, m\}.$$

En appliquant le Lemme 3.2.2, nous obtenons

$$S_\lambda^\infty = \{0\}.$$

2) **Unicité** : Comme f_μ est strictement convexe, il en résulte que la solution optimale de $(SDP)_\mu$ est unique.

3.2.2 Conditions d'optimalité pour $(SDP)_\mu$

Le problème $(SDP)_\mu$ est convexe, donc les conditions de K.K.T sont nécessaires et suffisantes. $X \in S_n^{++}$ est une solution optimale pour le problème $(SDP)_\mu$ si et seulement s'il existe $y \in \mathbb{R}^m$ tel que :

$$\begin{cases} C - \mu X^{-1} - \sum_{i=1}^m y_i A_i = 0 \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m. \end{cases} \quad (3.1)$$

Posons $Z = \mu X^{-1}$, le système (3.1) devient :

$$\begin{cases} \sum_{i=1}^m y_i A_i + Z = C, \quad Z \in S_n^{++} \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m, \quad X \in S_n^{++} \\ XZ = \mu I, \quad \mu > 0, \end{cases} \quad (3.2)$$

qui est le système paramétrisé du système (3.1). On note par $(X(\mu), y(\mu), Z(\mu))$ une solution du système (3.2). Nous savons déjà que $X(\mu)$ (et aussi $Z(\mu)$) est définie de façon unique. En supposant que les matrices A_i ($i = 1, \dots, m$) sont linéairement indépendantes, alors, $y(\mu)$ est aussi défini de manière unique.

Definition 1 *L'ensemble*

$$T_c = \{(X(\mu), y(\mu), Z(\mu)) \mid \mu > 0\},$$

est appelé la trajectoire centrale des problèmes SDP et SDD.

Théorème 3.2.2 [28] Si $(X(\mu), y(\mu), Z(\mu))$ est une solution optimale de $(SDP)_\mu$ avec $\mu > 0$, alors :

$$\lim_{\mu \rightarrow 0} (X(\mu), y(\mu), Z(\mu)) = (X^*, y^*, Z^*),$$

est une solution optimale de (SDP) .

Pour faciliter l'étude, nous considérons dans la suite la notation (X, y, Z) au lieu de $(X(\mu), y(\mu), Z(\mu))$.

3.3 Méthode de trajectoire centrale

3.3.1 Principe de la méthode de trajectoire centrale

La stratégie des méthodes de trajectoire centrale consiste à chercher des solutions approchées pour le système non linéaire (3.2) aux voisinage de la trajectoire centrale, i.e., en obtenant une suite décroissante de saut de dualité $X^k Z^k$ (ou d'une manière équivalente du paramètre μ qui tend vers zéro à la limite).

La méthode de Newton est considérée comme l'une des meilleures méthodes pour la résolution du système (3.2). Supposons à l'itération k , que $(X^k, y^k, Z^k) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$, on cherche un nouveau itéré $(X^{k+1}, y^{k+1}, Z^{k+1})$ défini par :

$$\begin{cases} X^{k+1} = X^k + \Delta X^k \\ y^{k+1} = y^k + \Delta y^k \\ Z^{k+1} = Z^k + \Delta Z^k. \end{cases}$$

Alors, $(X^{k+1}, y^{k+1}, Z^{k+1})$ satisfait le système non linéaire (3.2) et on a :

$$\begin{cases} \sum_{i=1}^m (y_i^k + \Delta y_i^k) A_i + (Z^k + \Delta Z^k) = C, & Z^k \in S_n^{++} \\ \langle A_i, X^k + \Delta X^k \rangle = b_i, & i = 1, \dots, m, X^k \in S_n^{++} \\ (X^k + \Delta X^k)(Z^k + \Delta Z^k) = \mu^k I, & \mu^k > 0, \end{cases} \quad (3.3)$$

où $(\Delta X^k, \Delta y^k, \Delta Z^k)$ est une solution du système linéaire :

$$\begin{cases} \sum_{i=1}^m \Delta y_i^k A_i + \Delta Z^k = 0, & Z^k \in S_n^{++} \\ \langle A_i, \Delta X^k \rangle = 0, & i = 1, \dots, m, X^k \in S_n^{++} \\ X^k \Delta Z^k + Z^k \Delta X^k = \mu^k I - X^k Z^k, & \mu^k > 0. \end{cases} \quad (3.4)$$

Pour mesurer la qualité de la solution trouvée, on introduit la notion de facteur de centralité ou de proximité défini par :

$$\delta(X, Z, \mu) = \|X^{1/2} Z X^{1/2} - \mu I\|.$$

La trajectoire centrale est donc, l'ensemble des points réalisables de proximité nulle pour un certain μ .

Un point est dit voisin de la trajectoire centrale s'il appartient à l'ensemble suivant :

$$\begin{aligned} S_\sigma(\mu) &= \{(X, y, Z) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ / \|X^{1/2} Z X^{1/2} - \mu I\| \leq \sigma \mu\} \\ &= \{(X, y, Z) \in T_c / \delta(X, Z, \mu) \leq \sigma \mu\}, \quad 0 < \sigma < 1. \end{aligned}$$

3.3.2 Algorithme de trajectoire centrale T_μ

Début algorithme

Initialisation :

Choisir $\varepsilon > 0$, $0 < \sigma < 1$ tel que $(X^0, y^0, Z^0) \in S_\sigma(\mu^0)$,
 $\mu^0 = \frac{\langle X^0, Z^0 \rangle}{n}$, $k = 0$.

Tant que $\mu^k > \varepsilon$ faire :

1. Calculer $(\Delta X^k, \Delta y^k, \Delta Z^k)$ en résolvant le système linéaire (3.4).
2. Prendre $(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k, y^k, Z^k) + (\Delta X^k, \Delta y^k, \Delta Z^k)$.
3. Prendre $\mu^{k+1} = \frac{\langle X^{k+1}, Z^{k+1} \rangle}{n}$ et $k = k + 1$.

Fin Tant que.

Fin algorithme.

Remarque 3.3.1 *La difficulté majeure des méthodes de trajectoire centrale est l'obtention d'une solution réalisable. Une fois que ce point est obtenu, la convergence de l'algorithme est garantie dès que ce point est au voisinage de la trajectoire centrale [10, 23, 29, 33, 40]. Dans ce sens, plusieurs études ont été menées pour répondre à cette question. En outre, de nombreux chercheurs ont démontré les résultats de convergence de l'algorithme, en introduisant le facteur poids dans la fonction barrière [13, 48].*

Signalons aussi que les directions du système (3.4) calculés par la méthode de Newton ne sont pas en général symétriques, ce qui pose le problème de faisabilité des itérés. Pour remédier, plusieurs familles de directions symétrisées sont proposées, à savoir direction (HAO) [3], direction (NT) [41], direction (HKM) [21, 28, 32] et la famille de direction de Monteiro [31, 32, 34].

Remarque 3.3.2 *L'équation non linéaire paramétrisée $XZ = \mu I$ du système (3.2), limite l'étude théorique et numérique de l'algorithme. Afin de donner plus de souplesse à l'équation $XZ = \mu I$, nous relaxons le paramètre μ par le rayon spectrale d'une matrice symétrique définie positive.*

3.4 Relaxation du paramètre μ

Nous proposons dans cette section, une généralisation de la méthode de trajectoire centrale primale-duale en remplaçant le paramètre μ dans le système (3.2) par le rayon spectral $\rho(W)$ de la matrice symétrique définie positive W . Le problème paramétrisé associé à SDP devient :

$$(SDP)_W \begin{cases} \min f_W(X) \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m, \end{cases}$$

où f_W est définie par :

$$f_W(X) = \begin{cases} \langle C, X \rangle - \rho(W) \ln \det X & \text{si } X \in S_n^{++} \\ +\infty & \text{sinon.} \end{cases}$$

Pour les mêmes raisons que dans $(SDP)_\mu$, le problème $(SDP)_W$ admet aussi une solution optimale unique (résultat du Théorème 3.2.1). Par conséquent, les conditions d'optimalité sont nécessaires et suffisantes. Il en résulte que X est une solution optimale de $(SDP)_W$ s'il existe $y \in \mathbb{R}^m$ de tel sorte que :

$$\begin{cases} C - \rho(W)X^{-1} - \sum_{i=1}^m y_i A_i = 0 \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m. \end{cases}$$

Posons $Z = \rho(W)X^{-1}$, alors on obtient :

$$\begin{cases} \sum_{i=1}^m y_i A_i + Z = C, & Z \in S_n^{++} \\ \langle A_i, X \rangle = b_i, & i = 1, \dots, m, X \in S_n^{++} \\ XZ = \rho(W)I, & W \in S_n^{++}. \end{cases} \quad (3.5)$$

On note par $(X(W), y(W), Z(W))$ une solution du système (3.5). Nous savons déjà que $X(W)$ (et aussi $Z(W)$) est définie de façon unique. En supposant que les matrices

A_i ($i = 1, \dots, m$) sont linéairement indépendantes, alors, $y(W)$ est aussi défini de façon unique.

L'ensemble des solutions $(X(W), y(W), Z(W))$ pour tout $W \succ 0$ forme la trajectoire centrale qui converge vers une solution optimale (X^*, y^*, Z^*) quand $\rho(W)$ tend vers 0.

Dans la suite, nous considérons (X, y, Z) au lieu $(X(W), y(W), Z(W))$ pour faciliter la présentation et l'étude.

Le système non linéaire (3.5) s'écrit comme :

$$F_W(X, y, Z) = 0, \text{ tel que } X, Z, W \in S_n^{++},$$

où

$$F_W(X, y, Z) = \begin{bmatrix} \sum_{i=1}^m y_i A_i + Z - C \\ \langle A_i, X \rangle - b_i, \quad i = 1, \dots, m \\ XZ - \rho(W)I. \end{bmatrix}.$$

En appliquant la méthode de Newton, la résolution du système non linéaire

$$F_W(X + \Delta X, y + \Delta y, Z + \Delta Z) = 0, \text{ tel que } X, Z, W \in S_n^{++},$$

revient à la résolution du système linéaire

$$F_W(X, y, Z) + (\Delta X, \Delta y, \Delta Z)^T \nabla F_W(X, y, Z) = 0, \text{ tel que } X, Z, W \in S_n^{++},$$

qui est équivalent au système linéaire suivant :

$$\begin{cases} \sum_{i=1}^m \Delta y_i A_i + \Delta Z = 0, & Z \in S_n^{++} \\ \langle A_i, \Delta X \rangle = 0, & i = 1, \dots, m, X \in S_n^{++} \\ X \Delta Z + Z \Delta X = \rho(W)I - XZ, & W \in S_n^{++}. \end{cases} \quad (3.6)$$

Par conséquent, la nouvelle itération est donnée par :

$$(X^+, y^+, Z^+) = (X, y, Z) + (\Delta X, \Delta y, \Delta Z).$$

3.4.1 Calcul de la direction

Contrairement à la programmation linéaire, où la direction primale-duale de Newton s'impose comme solution des équations linéarisées, il n'en va pas de même en optimisation (*SDP*). La difficulté vient de ce que XZ n'est pas nécessairement symétrique si l'équation $XZ = \mu I$ n'est pas vérifiée exactement, si bien que le système (3.4) a plus d'équations que d'inconnues. La question centrale est donc de savoir comment linéariser cette équation.

Une première possibilité serait de prendre comme troisième équation

$$-\mu X^{-1} + Z = 0.$$

Malheureusement, le calcul de l'inverse de X à chaque itération est coûteux. Pour cela, les algorithmes primaux-duaux de trajectoire centrale cherchent à symétriser l'équation $XZ = \mu I$, sans utiliser d'inverse des matrices. Zhang [47] a proposé une alternative pour garantir la symétrisation des directions comme suit :

On considère la transformation linéaire :

$$H_P(M) = \frac{1}{2} [PMP^{-1} + P^{-T}M^T P^T],$$

où P est une matrice inversible.

Zhang [47] a montré que : si M est une matrice définie positive, alors :

$$H_P(M) = \mu I \Leftrightarrow M = \mu I.$$

Donc, on peut remplacer la matrice XZ dans les systèmes (3.5) par $H_P(XZ)$.

- Si $P = I$, la direction obtenue coïncide avec la direction (AHO) de Alizadeh, Haeberly

et Overton [3].

- Si $P = [X^{\frac{1}{2}}(X^{\frac{1}{2}}ZX^{\frac{1}{2}})^{-\frac{1}{2}}X^{\frac{1}{2}}]^{\frac{1}{2}}$, on obtient la direction (NT) de Nesterov et Todd [41].
- Si $P = X^{-\frac{1}{2}}$ or $P = Z^{\frac{1}{2}}$, on obtient la famille des directions (HKM) de Helmberg et al, Kojima et al et Monteiro [21, 28, 32].

Grâce à sa simplicité en pratique, nous utilisons la direction (AHO). Pour cela, nous redéfinissons l'opérateur :

$$\begin{aligned} A & : S_n \rightarrow \mathbb{R}^m \\ A(X) & = (\langle A_i, X \rangle)_{i=1}^m. \end{aligned}$$

L'opérateur adjoint de A est défini par :

$$\begin{aligned} A^* & : \mathbb{R}^m \rightarrow S_n \\ A^*(y) & = \sum_{i=1}^m y_i A_i. \end{aligned}$$

En remplaçant XZ par $\frac{XZ + ZX}{2}$, le système (3.6) devient :

$$\begin{cases} A^*(\Delta y) + \Delta Z = 0, & Z \in S_n^{++} \\ A(\Delta X) = 0, & X \in S_n^{++} \\ \hbar(\Delta X) + \mathcal{L}(\Delta Z) = 2\rho(W)I - (XZ + ZX), & W \in S_n^{++}, \end{cases} \quad (3.7)$$

où

$$\begin{aligned} \hbar, \mathcal{L} & : S_n \rightarrow S_n \\ \hbar P & = \frac{1}{2}(ZP + PZ), \\ \mathcal{L} P & = \frac{1}{2}(XP + PX). \end{aligned} \quad (3.8)$$

Théorème 3.4.1 [48] *Soit $\hbar, \mathcal{L} : S_n \rightarrow S_n$ deux opérateurs, tel que \hbar est invertible et*

$\hbar^{-1}\mathcal{L}$ est un opérateur défini positif, alors la solution du système (3.7) est unique et donnée par :

$$\begin{cases} \Delta y = (A\hbar^{-1}\mathcal{L}A^*)^{-1}(-A\hbar^{-1}(2\rho(W)I - (XZ + ZX))) \\ \Delta Z = -A^*(\Delta y), \quad X, Z \in S_n^{++} \\ \Delta X = \hbar^{-1}(2\rho(W)I - (XZ + ZX) - \mathcal{L}(\Delta Z)), \quad W \in S_n^{++}. \end{cases} \quad (3.9)$$

Remarque 3.4.1 Le calcul de ΔX et ΔZ dans l'équations 2 et 3 du système (3.9) est basé sur le calcul de la valeur de Δy dans l'équation 1.

Calcul effectif de Δy

On peut écrire le système (3.9) sous la forme :

$$\begin{cases} A\hbar^{-1}\mathcal{L}A^*\Delta y = -A\hbar^{-1}(2\rho(W)I - (XZ + ZX)) \\ \Delta Z = -A^*(\Delta y), \quad X, Z \in S_n^{++} \\ \Delta X = \hbar^{-1}(2\rho(W)I - (XZ + ZX) - \mathcal{L}(\Delta Z)), \quad W \in S_n^{++}. \end{cases} \quad (3.10)$$

Posons $B = A\hbar^{-1}\mathcal{L}A^*$ une matrice de M_m telle que chaque élément de B est donné par :

$$B_{ij} = \langle A_i, \hbar^{-1}\mathcal{L}A_j \rangle, \quad i, j = 1, \dots, m.$$

On résoud le système linéaire :

$$\hbar U = K,$$

où $U \in S_n$ une matrice à calculer et $K \in S_n$ une matice donnée.

D'après (3.8)

$$\hbar U = \frac{1}{2}(ZU + UZ).$$

Ce qui est équivalent à

$$ZU + UZ = 2K. \quad (3.11)$$

Puisque $Z \in S_n^{++}$, et d'après la proposition 1.1.1, $Z = QDQ^t$, avec $Q^tQ = QQ^t = I$ et D est la matrice diagonale des valeurs propres de Z .

Donc l'équation (3.11) s'écrit :

$$D\bar{U} + \bar{U}Z = 2\bar{K},$$

avec :

$$\bar{U} = Q^tUQ \text{ et } \bar{K} = Q^tKQ,$$

les éléments de la matrice \bar{U} s'écrivent sous la forme :

$$\bar{U}_{ij} = 2\bar{K}_{ij}/d_i + d_j, \forall i, j = 1, \dots, j.$$

Finalement U sera obtenu en fonction de \bar{U} et on a :

$$U = Q\bar{U}Q^t.$$

Remarque 3.4.2 *La vitesse de convergence de l'algorithme dépend largement de la manière de calculer la direction $(\Delta X, \Delta y, \Delta Z)$ et le pas de déplacement le long de la direction. Ainsi, l'itération courante est définie par :*

$$\begin{cases} X^+ = X + \alpha\Delta X, \\ y^+ = y + \beta\Delta y, \\ Z^+ = Z + \beta\Delta Z. \end{cases}$$

3.4.2 Calcul du pas de déplacement

On cherche (α, β) qui garantisse la définie positivité des matrices X^+ et Z^+ , et améliore la vitesse de convergence de l'algorithme. Pour cela, nous procédons comme suit :

Trouver (α, β) de telle sorte que :

$$X^+ = X + \alpha \Delta X \in S_n^{++}.$$

$$Z^+ = Z + \beta \Delta Z \in S_n^{++}.$$

Nous considérons deux alternatives pour calculer (α, β) proposées par M. J. Todd, K. C. Toh et R. H. Tütüncü [41] et I. Touil, A. Keraghel et D. Benterki [42].

Alternative M. J. Todd, K. C. Toh et R. H. Tütüncü

$$\alpha = \min \left(1, \frac{-\tau}{\lambda_{\min}(X^{-1}\Delta X)} \right), \quad \beta = \min \left(1, \frac{-\tau}{\lambda_{\min}(Z^{-1}\Delta Z)} \right),$$

où, λ_{\min} est la plus petite valeur propre de $(X^{-1}\Delta X)$ (resp $(Z^{-1}\Delta Z)$) et τ est pris entre 0 et 1.

Alternative I. Touil, A. Keraghel et D. Benterki

$$\alpha_X = \begin{cases} \min \left(\frac{\sum_{i \neq j=1}^n |X_{ij}| - X_{ii}}{\Delta X_{ii} - \sum_{i \neq j=1}^n |\Delta X_{ij}|} \right) & \text{si } I_0 \neq \phi \\ 1 & \text{si } I_0 = \phi. \end{cases}$$

$$\beta_Z = \begin{cases} \min \left(\frac{\sum_{i \neq j=1}^n |Z_{ij}| - Z_{ii}}{\Delta Z_{ii} - \sum_{i \neq j=1}^n |\Delta Z_{ij}|} \right) & \text{si } I_1 \neq \phi \\ 1 & \text{si } I_1 = \phi, \end{cases}$$

où

$$I_0 = \left\{ i \in \{1, \dots, n\} / \Delta X_{ii} - \sum_{i \neq j=1}^n |\Delta X_{ij}| < 0 \right\}.$$

$$I_1 = \left\{ i \in \{1, \dots, n\} / \Delta Z_{ii} - \sum_{i \neq j=1}^n |\Delta Z_{ij}| < 0 \right\}.$$

Si on prend $\alpha^* = \min(\alpha_X, \beta_Z)$, alors :

$$X^+ = (X + \alpha^* \Delta X) \in S_n^{++} \text{ et } Z^+ = (Z + \alpha^* \Delta Z) \in S_n^{++}.$$

On dit que (X, y, Z) est au voisinage de la trajectoire centrale (noté $S_\sigma(W)$) s'il appartient à l'ensemble :

$$S_\sigma(W) = \left\{ (X, y, Z) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ / \left\| \frac{XZ + ZX}{2} - \rho(W)I \right\| \leq \sigma \rho(W) \right\},$$

avec $0 < \sigma < 1$.

Nous présentons maintenant la description générale de l'algorithme.

3.4.3 Algorithme de trajectoire centrale T_W

Début algorithme

Initialisation :

Choisir $\varepsilon > 0$ et $0 < \sigma < 1$ tel que $(X^0, y^0, Z^0) \in S_\sigma(W^0)$,

$$W^0 = \frac{\text{diag}(X^0 Z^0)}{n}, k = 0.$$

Tant que $\rho(W^k) > \varepsilon$ faire :

1. Calculer $(\Delta X^k, \Delta y^k, \Delta Z^k)$ en résolvant le système linéaire (3.7).
2. Prendre $(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k, y^k, Z^k) + \alpha^k (\Delta X^k, \Delta y^k, \Delta Z^k)$.

tel que $\alpha^k = \min(\alpha_{X^k}, \beta_{Z^k})$

3. Prendre $W^{k+1} = \frac{\text{diag}(X^{k+1} Z^{k+1})}{n}$ et $k = k + 1$.

Fin Tant que.

Fin algorithme.

3.4.4 Convergence de l'algorithme

Pour montrer le théorème de convergence, on a besoin du lemme suivant :

Lemme 3.4.1 [34] Soit $(X, y, Z) \in S_\sigma(W)$, et soit $(\Delta X, \Delta y, \Delta Z)$ la solution du système (3.9), alors :

$$X^+ Z^+ = (X + \alpha \Delta X)(Z + \alpha \Delta Z) = XZ + \alpha(\mu I - XZ).$$

Théorème 3.4.2 Soient $0 < \sigma < 1$, $\sigma' = (1 + \alpha)\sigma$. Supposons que $(X, y, Z) \in S_\sigma(W)$, et soit $(\Delta X, \Delta y, \Delta Z)$ solution du système (3.9), alors :

1. $(X^+, y^+, Z^+) = (X + \alpha \Delta X, y + \alpha \Delta y, Z + \alpha \Delta Z) \in S_{\sigma'}(W)$.
2. $\langle X^+, Z^+ \rangle = \langle X + \alpha \Delta X, Z + \alpha \Delta Z \rangle = (1 - \alpha(1 - \sigma)) \langle X, Z \rangle < \langle X, Z \rangle, \forall \alpha \in \mathbb{R}$.

Preuve :

1. On a $X^+ = X + \alpha \Delta X$, $Z^+ = Z + \alpha \Delta Z$, donc

$$\begin{aligned} \left\| \frac{X^+ Z^+ + Z^+ X^+}{2} - \rho(W)I \right\| &= \left\| \frac{(X + \alpha \Delta X)(Z + \alpha \Delta Z) + (Z + \alpha \Delta Z)(X + \alpha \Delta X)}{2} - \rho(W)I \right\| \\ &= \left\| \frac{XZ + ZX}{2} - \rho(W)I + \frac{\alpha}{2}(X\Delta Z + \Delta XZ + Z\Delta X + \Delta ZX) \right\| \\ &= \left\| \frac{XZ + ZX}{2} - \rho(W)I + \frac{\alpha}{2}(2\rho(W)I - (XZ + ZX)) \right\| \\ &= \left\| \frac{XZ + ZX}{2} - \rho(W)I + \alpha(\rho(W)I - \frac{XZ + ZX}{2}) \right\| \\ &= \left\| \frac{XZ + ZX}{2} - \rho(W)I \right\| + \alpha \left\| \rho(W)I - \frac{XZ + ZX}{2} \right\| \\ &\leq \sigma\rho(W) + \alpha\sigma\rho(W) \\ &= (1 + \alpha)\sigma\rho(W) = \sigma'\rho(W). \end{aligned}$$

D'où $(X^+, y^+, Z^+) \in S_{\sigma'}(W)$.

2.

$$\begin{aligned}\langle X^+, Z^+ \rangle &= \langle X + \alpha \Delta X, Z + \alpha \Delta Z \rangle \\ &= \langle X, Z \rangle + \alpha (\langle X, \Delta Z \rangle + \langle \Delta X, Z \rangle) + \alpha^2 \langle \Delta X, \Delta Z \rangle \\ &= \langle X, Z \rangle + \alpha (\langle X, \Delta Z \rangle + \langle \Delta X, Z \rangle) \\ &= \langle X, Z \rangle + \alpha \operatorname{tr}(\sigma \rho(W)I - (\frac{XZ + ZX}{2})) \\ &= \langle X, Z \rangle + \alpha \operatorname{tr}(\sigma \rho(W)I) - \alpha \operatorname{tr}(\frac{XZ + ZX}{2}) \\ &= \langle X, Z \rangle + \alpha \operatorname{tr}(\sigma \rho(W)I) - \alpha \operatorname{tr}(XZ) \\ &= \langle X, Z \rangle + \alpha n \sigma \rho(W) - \alpha \operatorname{tr}(XZ) \\ &= (1 - \alpha) \langle X, Z \rangle + \alpha n \sigma \rho(W) \\ &= (1 - \alpha) \langle X, Z \rangle + \alpha \sigma \langle X, Z \rangle \\ &= (1 - \alpha(1 - \sigma)) \langle X, Z \rangle < \langle X, Z \rangle.\end{aligned}$$

3.5 Tests Numériques

Nous présentons dans ce paragraphe quelques expérimentations numériques issues de la mise en œuvre des deux algorithmes T_μ et T_W décrits dans notre travail en utilisant le logiciel Matlab10. Les exemples testés sont pris de la littérature voir par exemple [8, 42]. Nous avons pris ϵ entre (10^{-5} et 10^{-7}), $\sigma = 0, 1$ et le pas de déplacement est calculé par l'alternative de I. Touil [42].

On note par **Iter**, le nombre des itérations nécessaires pour obtenir une solution optimale primale-duale (X^*, y^*, Z^*) et **Temps(s)**, le temps de calcul en seconde.

3.5.1 Exemples à taille fixe

Exemple 1 :

$n = 3, m = 2,$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

La solution primale-duale strictement réalisable initiale considérée est :

$$X^0 = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}, Z^0 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, y^0 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

La solution optimale primale-duale obtenue est :

$$X^* = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, Z^* = \begin{pmatrix} 1.4 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0 \end{pmatrix}, y^* = \begin{pmatrix} -0.4 \\ 0 \end{pmatrix}.$$

Exemple 2 :

$n = 5, m = 3,$

$$C = \begin{pmatrix} -4 & 0 & 0 & 0 & 0 \\ 0 & -5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 8 \\ 7 \\ 3 \end{pmatrix}$$

La solution primale-duale strictement réalisable initiale considérée est :

$$X^0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}, Z^0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}, y^0 = \begin{pmatrix} -2 \\ -1 \\ -2 \end{pmatrix}.$$

La solution optimale primale-duale obtenue est :

$$X^* = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, Z^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, y^* = \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix}.$$

Exemple 3 :

$n = 6, m = 3,$

$$C = \begin{pmatrix} -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 6 \\ 2 \\ 4 \end{pmatrix}.$$

La solution primale-duale strictement réalisable initiale considérée est :

$$X^0 = \begin{pmatrix} 1.467 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.086 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.36 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.26 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.086 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.532 \end{pmatrix},$$

$$Z^0 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, y^0 = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}.$$

La solution optimale primale-duale obtenue est :

$$X^* = \begin{pmatrix} 1.9945 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$S^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.2067 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.4139 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7926 \end{pmatrix}, y^* = \begin{pmatrix} 0 \\ -2.4130 \\ -0.7932 \end{pmatrix}.$$

Tableau des résultats comparatifs

ex (m, n)	paramètre μ		paramètre $\rho(W)$	
	Iter	Temps(s)	Iter	Temps(s)
ex (2, 3)	4	0.017895	3	0.011789
ex (3, 5)	7	0.018817	5	0.012493
ex (3, 6)	7	0.022879	6	0.018049

3.5.2 Exemples à taille variable

Rappelons que le problème considéré est

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

où son dual est défini comme suit

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

avec

$C = -I$, $b[i] = 2$, $i = 1, \dots, m$, $n = 2m$, et les matrices A_k , $k = 1, \dots, m$ sont définies

comme suit :

$$A_k[i, j] = \begin{cases} 1 & \text{si } i = j = k, \\ 1 & \text{si } i = j \text{ et } i = m + k, \\ 0 & \text{sinon.} \end{cases}$$

La solution primale-duale strictement réalisable initiale considérée est :

$$\begin{aligned}
 X^0 [i, j] &= \begin{cases} 1,5 & \text{si } i = j = 1, \dots, m, \\ 0,5 & \text{si } i = j = m + 1, \dots, m, \\ 0 & \text{ailleurs.} \end{cases} \\
 Z^0 [i, j] &= \begin{cases} 1 & \text{si } i = j = 1, \dots, n, \\ 0 & \text{ailleurs.} \end{cases} \\
 y^0 [i] &= -2, \quad i = 1, \dots, m.
 \end{aligned}$$

Tableau des résultats comparatifs

ex (m, n)	paramètre μ		paramètre $\rho(W)$	
	Iter	Temps(s)	Iter	Temps(s)
(10, 20)	7	0.031858	3	0.029764
(30, 60)	8	0.163532	3	0.090465
(50, 100)	8	0.458340	3	0.181330
(100, 200)	8	1.579016	3	0.640606
(200, 400)	8	7.010987	3	2.837008

3.5.3 Commentaires

À travers les tests numériques qu'on a réalisé sur des exemples de différentes dimensions, on remarque que les résultats montrent l'importance de la modification introduite sur le paramètre barrière, exprimée par une réduction significative du nombre d'itérations et du temps de calcul. En général, le choix du paramètre $\rho(W)$ est meilleur que celui du paramètre μ .

Conclusion

Dans cette thèse, nous avons étudié une méthode primale-duale de type trajectoire centrale basée sur la paramétrisation de l'objectif du problème de la programmation semi-définie.

En effet, nous avons montré l'existence et l'unicité de la solution optimale du problème paramétrisé, puis nous avons mis l'accent sur le comportement de l'algorithme correspondant. À cet égard, nous avons introduit un nouveau paramètre (barrière) afin de relaxer l'étude théorique et numérique de cet algorithme. Les résultats obtenus sont encourageants et ouvrent la voie à d'autres modifications qui visent à obtenir des algorithmes plus efficaces en terme de convergence.

Bibliographie

- [1] F. Alizadeh, Combinatorial Optimization with Interior-Point Methods and Semi-Definite Matrices, Ph.D. Thesis, Computer Science Department, University of Minnesota, Minneapolis, MN, 1991.
- [2] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, *SIAM J. Optim.* 5 (1995) 13-51.
- [3] F. Alizadeh, J. P. A. Haeberly & M. L. Overton, Primal-dual interior-point methods for semidefinite programming : Convergence rates, stability and numerical results, *SIAM J. Optim.* 8 (1998) 746–768.
- [4] E. D. Andersen and K. D. Andersen, The MOSEK interior point optimizer for linear programming : an implementation of the homogeneous algorithm, In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, High performance optimization, pages 197–232. Kluwer Academic Publishers, 2000.
- [5] 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, 2002.
- [6] Y. Q. Bai, M. El. Ghani, C. Roos, A new efficient large-update primal-dual interior point method based on a finite barrier, *SIAM J. Optim.* 13 (3) (2003) 766–782 .
- [7] R. Bellman and K. Fan, On systems of linear inequalities in hermitian matrix variables, In V.L. Klee, editor, Convexity, Vol. 7 Proc. Symposia in Pure Mathematics, pages 1–11. Amer. Math. Soc., Providence, RI, 1963.

- [8] D. Benterki, J. P. Crouzeix, B. Merikhi, A numerical implementation of an interior point method for semidefinite programming, *Pesquisa Operacional* 23, no 1 (2003) 49–59.
- [9] D. Benterki, Résolution des problème de programmation semi-définie par des méthode de réduction du potentiel, thèse de doctorat, département de mathématiques, Université Ferhat Abbas, Sétif1, Algérie 2004.
- [10] N. Brixius, F. A. Porta and R. Sheng, Nonsymmetric search directions for semidefinite programming, *SIAM J. Optim.* (1999) 863–876.
- [11] G. M. Cho, An interior point algorithm for linear optimization based on a new barrier function, *Appl. Math. Comput.* 218 (2011) 386–395.
- [12] 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 (2009) 2628–2640.
- [13] C. B. Chua, A new notion of weighted centers for semidefinite programming, *SIAM J. Optim.* 16 (2006) 1092–1109.
- [14] B. Craven and B. Mond, Linear programming with matrix variables, *Linear Algebra Appl.* 38 (1981) 73–80.
- [15] E. Djeflal, Etude numérique d’une méthode de la trajectoire centrale pour la programmation convexe sous contraintes linéaires, thèse de Magister, Université de Batna, 2009.
- [16] J. Ding and T. Y. Li, An algorithm based on weighted logarithmic barrier functions for linear complementarity problems, *Arabian journal for science and engineering* 15 (1990) 679–687.
- [17] M. El Ghami, I. D. Ivanov, C. Roos, T. Steihaug, A polynomial-time algorithm for LO based on generalized logarithmic barrier functions, *Int. J. Appl. Math.* 21 (2008) 99 –115.

- [18] M. El Ghami, C. Roos and T. Steihauga, A generic primal–dual interior-point method for semidefinite optimization based on a new class of kernel functions, *Optimization Methods & Software*. 25 (3) (2010) 387–403.
- [19] C. C. Gonzaga, Path following methods for linear programming, *SIAM Review*. 34 (1992) 167–227.
- [20] M. Halicka, E. De Klerk and C. Roos, On the convergence of the central path in semidefinite optimization, *SIAM J. Optim.* 12 (2002) 1090–1099.
- [21] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz, An interior-point method for semidefinite programming, *SIAM J. Optim.* 6 (1996) 342–361.
- [22] J. Ji, F. A. Potra and R. Sheng, On the local convergence of a predictor-corrector method for semidefinite programming, *SIAM J. Optim.* 10 (1999) 195–210.
- [23] C. Kanzow and C. Nagel, Semidefinite programs : New search directions, smoothing-typemethods, and numerical results, *SIAM J. Optim.* 13 (2002) 1–23.
- [24] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica*, 4 (1984) 373–395.
- [25] Z. Kebbiche, Etude et extensions d’algorithmes de points intérieurs pour la programmation non linéaire, thèse de doctorat d’état, Université Ferhat Abbas Setif 1 2007.
- [26] A. Keraghel, *Analyse convexe : Théorie fondamentale et exercices*, Editions Dar El’Houda, Ain M’lila, Algérie, 2001.
- [27] S. Kettab, D. Benterki, A relaxed logarithmic barrier method for semidefinite programming, *RAIRO-Oper. Res.* 49 (2015) 555–568.
- [28] M. Kojima, S. Shindoh, and S. Hara, Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices, *SIAM J. Optim.* 7 (1997) 86–125.

- [29] M. Kojima, M. Shida and S. Shindoh, A predictor-corrector interior-point algorithm for the semidefinite linear complementarity problem using the Alizadeh-Haeberly-Overton search direction, *SIAM J. Optim.* 9 (1999) 444–465.
- [30] I. J. Lustig, R. E. Marsten, and D. F. Shanno, Interior point methods : Computational state of the art, *ORSA Journal on Computing*, 6 (1994) 1–15.
- [31] R. D. C. Monteiro and T. Tsuchiya, Polynomial convergence of a new family of primal-dual algorithms for semidefinite programming, *SIAM J. Optim.* 9 (1999) 551–577.
- [32] R. D. C. Monteiro, Primal-dual path-following algorithms for semidefinite programming, *SIAM J. Optim.* 7 (1997) 663–678.
- [33] R. D. C. Monteiro and P. R. Zanjacomo, A note on the existence of the Alizadeh-Haeberly-Overton direction for semidefinite programming, *School of Industrial and Systems Engineering, Georgia Institute of Technology Atlanta*, 78 (1997) 393–396.
- [34] R. D. C. Monteiro, Polynomial convergence of primal-dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions, *SIAM J. Optim.* 8 (1998) 797–812.
- [35] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming : Theory and Applications*, SIAM, Philadelphia, PA, 1994.
- [36] A. M. Nunez and R. M. Freund, Condition-measure bounds on the behavior of the central trajectory of semidefinite program. *SIAM J. Optim.* 12 (2001) 818–836.
- [37] J. Peng, C. Roos, T. Terlaky, Self-regularity functions and new search directions for linear and semidefinite optimization, *Math. Prog.* 93 (2002) 129-171.
- [38] M. R. Peyghami, S. F. Hafshejani, Complexity analysis of an interior point algorithm for linear optimization based on a new proximity function, *Numer. Algo.* 67 (2014) 33–48.
- [39] C. Roos, T. Terlaky, and J. Ph. Vial, *Theory and Algorithms for Linear Optimization : An interior point approach*, John Wiley & Sons, New York, 1997.

- [40] M. J. Todd, On search directions in interior-point methods for semidefinite programming, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 1997.
- [41] M. J. Todd, K. C. Toh and R. H. Tütüncü, On the Nesterov-Todd direction in semidefinite programming, *SIAM J. Optim.* 8 (1998) 769–796.
- [42] I. Touil, Etude comparative des performances d’une méthode de points intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire, thèse de magister, département de Mathématique, Université Ferhat Abbas, Sétif 1, Algérie 2005.
- [43] G. Q. Wang, Y. Q. Bai, and C. Roos, Primal-Dual Interior-Point Algorithms for Semidefinite Optimization Based on a Simple Kernel Function, *Journal of Mathematical Modelling and Algorithms* 4 (2005) 409–433.
- [44] G. Q. Wang, Y. Q. Bai, A new primal-dual path-following interior-point algorithm for semidefinite optimization, *J. Math. Anal. Appl.* 353 (2009) 339–349.
- [45] S. J. Wright, Primal–dual interior point methods, SIAM, Philadelphia, 1997.
- [46] Y. Ye, Interior point algorithms, *Discrete Mathematics and Optimization*. Wiley-Interscience, New York, 1997.
- [47] Y. Zhang, On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming, *SIAM J. Optim.* 8 (1998) 365–386.
- [48] L. Zhaosong, Algorithm design and analysis for large-scale semidefinite programming and nonlinear programming, thesis presented to the academic faculty in partial Fulfillment of the requirements for the degree doctor of philosophy, School of Industrial and Systems Engineering Georgia Institute of Technology, 2005.

ملخص

تعرف طرق النقطة الداخلية الأولية-المرادفة بأنها الأكثر فعالية في حل مسائل البرمجة المثالية كبيرة النمط، مثل مسائل البرمجة الخطية، مسائل البرمجة التربيعية، مسائل البرمجة نصف المعرفة و مسائل البرمجة المحدبة. كما تتميز هذه الطرق بتقاربها الحدودي و بسلوكها العددي الجيد.

نقترح في دراستنا طريقة جديدة من نوع المسار المركزي للبرمجة الخطية نصف معرفة حيث أننا قمنا بإرخاء الوسيط من أجل إعطاء ليونة سواء في الجانب النظري أو الجانب العددي وتسريع تقارب الخوارزمية. وهذا ما تؤكدته التجارب العددية التي تبين السلوك الجيد للخوارزمية المقترحة.

كلمات مفاتيح : البرمجة الخطية، البرمجة الخطية نصف معرفة موجبة، طريقة المسار المركزي.

Abstract

Primal and dual interior-point methods (IPMs) have been well known as the most effective methods for solving wide classes of optimization problems, such that, linear optimization problem, quadratic optimization problem, semidefinite optimization problem and convex optimization problem. These methods have a polynomial convergence and are credited of a good numerical behaviour. In our study, we propose a new central trajectory method for linear semidefinite programming where a relaxation of barrier parameter is introduced in order to give more flexibility to the theoretical and numerical aspects of the perturbed problems and accelerate the convergence of the algorithm. These purposes are confirmed by numerical tests showing the good behaviour of the proposed algorithm.

Key words: Linear Programming, Linear Semidefinite Programming, central trajectory Method.

Résumé

Les méthodes primales-duales de points intérieurs ont été bien connues les plus efficaces pour résoudre les classes de large taille de problèmes d'optimisation, tel que, problème d'optimisation linéaire, problème d'optimisation quadratique, problème d'optimisation semi-définie et problème d'optimisation convexe. Ces méthodes possèdent une convergence polynomiale et sont crédités d'un bon comportement numérique. Dans notre étude, nous proposons une nouvelle méthode de trajectoire centrale primale-duale pour la programmation semi-définie linéaire, où on a introduit une relaxation du paramètre barrière afin de donner plus de flexibilité aux aspects théoriques et numériques des problèmes perturbés, et d'accélérer la convergence de l'algorithme. Ces propos sont confirmés par des tests numériques montrant le bon comportement de l'algorithme proposé.

Mots clés : Programmation Linéaire, Programmation Semi-Définie Linéaire, Méthode de Trajectoire centrale.