



Université Batna 2 – Mostefa Ben Boulaïd
Faculté des Mathématiques et Informatique
Département de Mathématiques



Thèse

Présentée en vue de l'obtention du Diplôme de

DOCTORAT EN SCIENCES

Option : Mathématiques appliquées

Présentée par :

Guemmaz Abderrahim

Thème :

Problème de complémentarité semi définie

Basé sur Quelques fonctions noyaux

Soutenue le : 12/07/2022

Devant le jury composé de :

Dr. Zouhir Mokhtari	Prof.	Université de Batna 2	Président
Dr. El Amir Djefal	Prof.	Université de Batna 2	Rapporteur
Dr. Yahia Djabran	Prof.	Université de Biskra	Examineur
Dr. Lekhdari Imed	Prof.	Université de Biskra	Examineur
Dr. Brahimi Mahmoud	Prof.	Université de Batna 2	Invité

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de Batna 2
Faculté des Mathématiques et de l'Informatique
Département de Mathématiques

Thèse

Présentée en vue de l'obtention du Diplôme de

DOCTORAT EN SCIENCES

Option : Mathématiques appliquées

Par

Guemmaz Abderrahim

Thème

**Problème de complémentarité semi définie basé
sur quelques fonctions noyaux**

Soutenue le :

Devant le jury

Président :	Dr. Zouhir Mokhtari	Pr. Université de Batna 2
Rapporteur :	Dr. El Amir Djefal	Pr. Université de Batna 2
Examineurs :	Dr. Yahia Djabran	Pr. Université de Biskra
	Dr. Lekhdari Imed	Pr. Université de Biskra
	Dr. Brahimi Mahmoud	Pr. Université de Batna 2

Remerciements

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

Je remercie chaleureusement mes très chers parents et ma belle-mère qui ont toujours été là pour moi, et pour leurs soutiens qui m'a été bien utile durant ma thèse.

Je tiens à exprimer ma profonde reconnaissance envers mon directeur de thèse Monsieur : El **Amir DJEFFAL**, Professeur à l'Université Batna-2, pour sa disponibilité, ses conseils et ses encouragements qui m'ont été très estimables, son aide et discussion et son support pendant l'élaboration de cette thèse.

Je remercie tout particulièrement Monsieur **Zouhir MOUKHTARI** Professeur à l'Université de Batna -2, pour l'honneur qu'il m'a fait en acceptant de présider le jury de cette thèse. Je suis très honoré par la présence dans le jury des Messieurs : **Yahia DJABRAN** Professeur à l'Université de Biskra, **Mahmoud BRAHIMI** Professeur à l'Université de Batna -2, et **Imed LEKHDARI** Professeur à l'Université de Biskra, qui ont bien voulu être juges ce travail. Je leur adresse toute ma reconnaissance.

Je remercie mes chers et adorable sœurs et frères pour le soutien et pour l'encouragement.

A titre plus personnel, Je remercie chaleureusement ma chère femme pour sa grande patience, sa compréhension et sacrifice, l'encouragement ininterrompu et ses nombreux conseils tout au long de ma thèse.

Un grand merci à mes Beaux amours et la prunelle de mes yeux "Mes enfants" que Dieu les protège.

Mes sincères remerciements s'adressent également à mes amis : Dr. Bachir BOUNIBANE, Dr. Fateh MERAHI, et Dr. Souhaib MILLES .

Abderrahim Guemmaz

Résumé

Dans cette thèse, nous étudions la méthode de points intérieurs (MPI) de trajectoire centrale de type primal-dual basée sur une fonction noyau pour résoudre des problèmes d'optimisation semi-définie (OSD). Pour cette étude, nous proposons une nouvelle fonction noyau avec un terme de barrière logarithmique. Nous proposons aussi des directions de recherche, et une fonction de proximité qui sont basées sur cette fonction noyau, et on donne la complexité des algorithmes à grand et à petit pas. L'efficacité numérique de ces algorithmes est confirmée par des tests numériques qui sont encouragées.

Mots clés: Optimisation semi-définie, Méthode de points intérieurs, Chemin centrale, Fonction noyau, Complexité algorithmique.

Abstract:

In this thesis, we study a primal-dual Interior Point Method (IPMs) of central path type based on a new kernel function for solving the semidefinite problems (SDO). For this study, we propose a new kernel function with a logarithmic barrier term. We also propose a search directions, and a proximity function which are based on this kernel function, and we give the complexity of the large and small update algorithms. The numerical efficiency of these algorithms is confirmed by some numerical tests which are encouraged.

Keywords: Semidefinite optimisation, Interior point method, Cental path, Kernel function, Algorithmic complexity.

ملخص:

في هذه الأطروحة، ندرس طريقة النقاط الداخلية (IPM) الأولية-المرادفة من نوع المسار المركزي التي تعتمد على دالة نواة لحل مسائل التحسين نصف المعرف (SDO). لهذه الدراسة نقترح دالة نواة جديدة ذات مصطلح حاجز لوغاريتمي نقترح أيضا اتجاهات البحث، ودالة القرب والتي تعتمد على دالة النواة هذه، ونعطي تكلفة خوارزميات ذات خطوة كبيرة وصغيرة. تم تأكيد الكفاءة العددية لهذه الخوارزميات من خلال الاختبارات العددية وهي مشجعة.

الكلمات المفتاحية: التحسين نصف المعرف، طريقة النقاط الداخلية، المسار المركزي، دالة النواة، تكلفة الخوارزمية.

Table des matières

Introduction	4
Notations et terminologie	9
1 Préliminaires et notions fondamentales	12
1.1 Quelques notions sur les matrices	12
1.1.1 Valeurs propres et spectre	12
1.1.2 Trace d'une matrice	14
1.1.3 Produit scalaire et normes	14
1.1.4 Matrices semi-définies positives	16
1.2 Rappel d'analyse convexe	18
1.2.1 Ensembles affines	18
1.2.2 Fonctions affines	19
1.2.3 Ensembles convexes	19
1.2.4 Fonctions convexes	20
1.2.5 Fonction mid-convexe	20
1.2.6 Caractérisation d'une fonction convexe différentiable	20
1.2.7 Fonction barrière	21
1.2.8 Cônes	22
1.3 Programmation mathématique	22
1.3.1 Définitions	23

Table des matières

1.3.2	Classification	24
1.3.3	Résolution d'un programme mathématique (PM)	25
1.3.4	Dualité lagrangienne	26
1.3.5	Méthode de Newton pour résoudre un système non linéaire	27
1.3.6	Méthodes de directions admissibles	28
1.3.7	Méthode de résolution d'un programme mathématique	28
2	Programmation semi-définie	31
2.1	Formulation du problème	31
2.2	Cas particuliers	32
2.2.1	Programmation linéaire	32
2.2.2	Programmation quadratique	33
2.3	Domaines d'application	35
2.3.1	Optimisation des valeurs propres	35
2.3.2	Programmation quadratique avec des contraintes quadratiques	37
2.3.3	Problème de programmation non linéaire	38
2.4	Dualité en programmation semi-définie	40
2.4.1	Relations primales-duales pour la programmation semi-définie	42
2.4.2	Dualité faible	42
2.4.3	Dualité forte	45
2.5	Complémentarité en (SDP)	45
2.6	Méthodes de résolution de (SDP)	46
2.6.1	Méthodes de points intérieurs	47
3	Algorithme de points intérieurs pour un problème d'optimisation semi défini	
	(SDO) basé sur une nouvelle fonction noyau	52
3.1	Description de l'algorithme	53
3.1.1	Méthode de la trajectoire centrale	53
3.1.2	Direction de descente	55
3.1.3	Algorithme générique de points intérieurs pour (SDO)	56

Table des matières

3.2	Fonction noyau et ses propriétés	58
3.3	Convergence de l'algorithme	63
3.3.1	Le pas de déplacement	64
3.3.2	Diminution de la fonction de proximité	73
3.3.3	Les bornes d'itération	75
3.4	Tests numériques	77
	Conclusion	79
	Bibliographie	80

Introduction

La programmation mathématique, branche de l'optimisation, s'occupe de la minimisation (ou la maximisation) sous contraintes d'une fonction à plusieurs variables, schéma très général s'appliquant à de nombreuses situations pratiques dans beaucoup de domaines (minimisation de coûts, de durées, maximisation de gains, de rentabilité, etc.).

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 [41]). Un article sur la programmation semi-définie paru en 1981 intitulé "Linear Programming with Matrix Variables" (Craven et Mond [4]), 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 :

$$\min \{Tr(CX) : Tr(A_i X) = b_i, i = 1, \dots, m, X \succeq 0\},$$

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. Ce problème est d'une importance capitale pour au moins trois raisons : la première est que la SDP contient des classes importantes de problèmes comme par exemple les cas spéciaux des programmations linéaire et quadratique. La deuxième raison est que ce problème a des applications en optimisation combinatoire, en théorie d'approximation, on système et théorie de contrôle, en mécanique, en ingénierie, en électricité, en économie, ect. La

dernière est que le problème SDP peut être résolu en temps polynomial par des algorithmes de points intérieurs.

Les algorithmes de points intérieurs de la SDP ont été étudiés intensivement dans les années 90, et c'est ce qui explique la résurgence de l'intérêt de la recherche en programmation semi-définie. En considérant les liens entre la programmation linéaire PL et la programmation semi-définie, on peut se demander comment les algorithmes de points intérieurs de la PL ont été étendus avec succès à la SDP ?

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 [39] 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.

La plupart des idées sous-jacentes à les méthodes de points intérieurs proviennent du domaine de la programmation non linéaire. Parmi leurs avantages, citons :

1. Efficacité théorique : ces méthodes ont une complexité polynomiale et s'exécutent en temps polynomial.
2. Efficacité pratique : les temps de calcul et de réponse de ces méthodes sont compétitifs.
3. Traitement de très grands problèmes : ces méthodes permettent de résoudre des problèmes de très grande taille en un temps acceptable et raisonnable.

4. Adaptation au cas non linéaire : il est possible d'adapter ces méthodes à la programmation non linéaire, plus particulièrement à la programmation convexe, ce qui permet de traiter de nouveaux types de problèmes pour lesquels on ne connaissait jusqu'à présent aucune méthode de résolution efficace.

On désigne par méthode de points intérieurs (MPI), toute procédure de résolution générant une suite de points appartenant à l'intérieur (relatif) du domaine réalisable (admissible) et convergeant vers une solution optimale du programme considéré. Il y a principalement trois grandes catégories de méthodes de points intérieurs :

- Les méthodes affines.
- Les méthodes de réduction du potentiel.
- Les méthodes de trajectoire centrale.

On peut distinguer différents MPI, selon qu'il s'agit des MPI réalisables ou non-réalisables. Les MPI réalisables commencent par une matrice strictement réalisable et conservent la faisabilité pendant le processus de résolution. Cependant, pour obtenir un tel point, il est généralement aussi difficile que de résoudre le problème sous-jacent. Les MPI non-réalisables pour les problèmes de programmation semi-définis commencent par une matrice définie-positive arbitraire. La performance des MPI non réalisables existantes dépend fortement du choix du point de départ.

Récemment, Kojima, Shindoh et Hara [30], Monteiro [45] et Y. Zhang [52] 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 [32, 5, 21, 47, 46, 2].

Les fonctions noyaux jouent un rôle important dans la conception de nouveaux algorithmes de points intérieurs de type primal-dual. Les méthodes de points intérieurs basées sur la notion des fonctions noyaux ont fait récemment l'objet de plusieurs études citons, [56, 12, 55, 49, 33, 36, 35].

Premièrement, Peng et all. [24] ont présenté des méthodes de points intérieurs pour (PL) et (SDP) basées sur les fonctions barrières auto-régulière, puis, Bai et all. [56, 55, 57], ont proposé une classe des méthodes de points intérieurs de type primal-dual pour (PL) basé sur une variété de fonctions noyaux non-auto-régulière et ils ont obtenu la même complexité favorable pour les algorithmes à grand et petit-pas que [24]. De plus, Wang et all ont fait l'extension des résultats ci-dessus obtenus pour (PL) à (SDP) [16] et ($CQSDP$) [14].

Pour d'autre algorithmes de points intérieurs basé sur les fonctions noyaux on cite [54, 1, 15, 16, 37, 9]. Notons que la meilleure borne d'itérations pour les algorithmes à petit et à grand pas, $O(\sqrt{n}(\log \frac{n}{\varepsilon}))$ et $O(\sqrt{n} \log n(\log \frac{n}{\varepsilon}))$, respectivement.

Comme les fonctions noyaux sont utilisées dans des algorithmes du chemin central, nous nous sommes intéressés à une méthode de trajectoire centrale de type primal-dual résoudre des problèmes de la programmation semi-définie basée sur les fonctions noyaux qui sont satisfont quelques propriétés et qui conduisent à un modèle très important pour développer ces méthodes. Ces méthodes déterminent une nouvelle classe de directions de Newton et sont considérées comme une proximité pour mesurer la qualité des points par rapport au chemin central.

Notre travail apporte sur des contributions théoriques, algorithmiques et numériques pour résoudre convenablement des problèmes d'optimisation de la programmation semi-définie. Nous proposons une méthode de résolution de la programmation semi-définie (SDP) par la méthode de trajectoire centrale, via une nouvelle fonction noyau avec un terme de barrière logarithmique [3] définie par :

$$\psi_S(t) = \frac{(t^2 - 1)}{2} - \frac{\log(t)}{2} - \frac{1}{2S} \sum_{j=1}^S \frac{t^{1-jq} - 1}{1 - jq}, q > 1, S \in \mathbb{N} \setminus \{0\}.$$

Nous avons montré que l'algorithme correspondant converge après $O(qSn^{\frac{Sg+1}{2Sg}} \log \frac{Tr(X^0 S^0)}{\epsilon})$ itérations pour les méthodes à grand-pas, et $O(q^2 S^2 \sqrt{n} \log \frac{Tr(X^0 S^0)}{\epsilon})$ itérations pour les méthodes à court-pas.

La thèse se présente comme suit :

Le **chapitre 1** présente un rappel sur des notions fondamentales d'usage fréquent pour la suite, à savoir : L'analyse convexe, quelques résultats de programmation mathématique et les notions des principales propriétés des cônes des matrices symétriques semi-définies positives. Dans le **chapitre 2**, nous l'avons consacré à la programmation semi-définie, on donne alors la formulation générale de ce type de problème et sa dualité. On précise ainsi quelques domaines d'application et on termine par les méthodes de résolution d'un problème (SDP). Dans le dernier chapitre c'est-à-dire le **chapitre 3**, nous proposons une nouvelle fonction de proximité pour un problème d'optimisation semi défini SDO par une nouvelle fonction noyau. En outre, nous formulons un algorithme de points intérieurs primal-dual pour (SDO) en utilisant la fonction de proximité et de donner son analyse de la complexité.

Enfin, nous clôturerons ce travail par une conclusion et perspectives suivi par Annexe et Bibliographie.

Notations et terminologie

Terminologie

- $(IPMs)$: Les méthodes de points intérieurs,
- (SDO) : Optimisation semi-définie,
- (SDP) : Programmation semi-définie,
- $(DSDP)$: Ledual de (SDP) ,
- (PL) : Programmation linéaire,
- (PM) : Programmation mathématique,
- (CQP) : Programmation quadratique convexe,
- $(CQSDP)$: Programme quadratique convexe esemi-définie,
- (P) : Le problème primal,
- (D) : Le problème dual,
- $(K.K.T)$: Karush-Kuhn-Tucker,
- $c - \grave{a} - d$: C'est à dire,
- NT : Nesterov et Todd,
- (CPI) : Condition de points intérieurs,
- $:=$: Egale par définition,
- $resp$: Respectivement,

Notations

\mathbb{R}^n	:	L'espace euclidien des n -composantes réelles,
\mathbb{R}_+^n	:	L'orthant positif de l'espace \mathbb{R}^n ,
$\mathbb{R}^{n \times m}$:	L'espace des matrices réelles à m lignes et n colonnes,
$\mathbb{R}^{n \times n}$:	L'espace de smatrices carrées d'ordre n ,
\mathbb{S}^n	=	$\{X : X \in \mathbb{R}^{n \times n}, X = X^T\}$,
\mathbb{S}_+^n	=	$\{X : X \in \mathbb{S}^n, X \succeq 0\}$,
\mathbb{S}_{++}^n	=	$\{X : X \in \mathbb{S}^n, X \succ 0\}$,
$\text{int}(D)$:	L'intérieur du domaine D ,
$\nabla f(x)$	=	$(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \dots, \frac{\partial f}{\partial x_n}(x))$, gradient de f au point x ,
$\nabla^2 f(x)$	=	$(\frac{\partial^2 f}{\partial x_i \partial x_j}(x))_{1 \leq i, j \leq n}$, la matrice Hessienne de f ,
$\frac{\partial f}{\partial x_i}(x)$:	Les dérivées partielles au point x ,
A^T	:	La matrice transposée de A ,
A^{-1}	:	L'inverse d'une matrice régulière A ,
$A^{\frac{1}{2}}$:	La racine carrée de la matrice $A \succeq 0$,
$A \sim B$:	Il existe une matrice inversible P telle que $A = PBP^{-1}$, (semblable),
$\text{sp}(A)$:	Le spectre de la matrice A ,
$\lambda_i(A)$:	La i^{eme} valeur propre de $A \in \mathbb{R}^{n \times n}$,
$\text{Tr}(A)$	=	$\sum_i a_{ii} = \sum_i \lambda_i(A)$, la trace d'une matrice $A \in \mathbb{R}^{n \times n}$,
$\rho(A)$	=	$\max_i \lambda_i(A) $, le rayon spectral de A ,
$\lambda_{\max}(A)$	=	$\max_i \lambda_i(A)$,
$\lambda_{\min}(A)$	=	$\min_i \lambda_i(A)$,
$\det(A)$	=	$\prod_i \lambda_i(A)$, déterminant de $A \in \mathbb{R}^{n \times n}$,
$A \succeq 0$:	A est une matrice symétrique semi-définie positive,
$A \succ 0$:	A est une matrice symétrique définie positive,

Notations et terminologie

I	:	La matrice identité d'ordre n ,
$A \bullet B$	=	$Tr(A^T B), \forall A, B \in \mathbb{R}^{n \times n}$,
X^k	:	La k^{eme} terme d'une suite de matrices,
$diag(x)$	=	X , la matrice diagonale avec $X_{ii} = x_i$,
$x > 0$:	Les composantes de $x, x_i > 0$ pour tout i ,
$x \geq 0$:	Les composantes de $x, x_i \geq 0$ pour tout i ,
x^T	:	Transposé du vecteur x de \mathbb{R}^n ,
x_i	:	La i^{eme} composante de x ,
e	:	Un vecteur de \mathbb{R}^n tel que $e_i = 1, \forall i = 1, \dots, n$,
x^k	:	La k^{eme} vecteur d'une suite de vecteurs,
xs	=	$(x_1 s_1, \dots, x_n s_n)^T$, produit d'Hadamard,
$\frac{x}{s}$	=	$(\frac{x_1}{s_1}, \frac{x_2}{s_2}, \dots, \frac{x_n}{s_n})^T : (s \neq 0)$,
x^{-1}	=	$(\frac{1}{x_1}, \frac{1}{x_2}, \dots, \frac{1}{x_n})^T : (x \neq 0)$,
\sqrt{x}	=	$(\sqrt{x_1}, \dots, \sqrt{x_n})^T : (x \geq 0)$,
$\langle x, s \rangle$	=	$x^T s = \sum_{k=1}^n x_k s_k$: le produit scalaire de deux vecteurs,
$\ A\ _F$	=	$\sqrt{Tr(A^T A)} = \sqrt{\sum_i \sum_j a_{ij}^2}$,
$\ A\ _2$	=	$\sqrt{\rho(A^T A)}$, la norme spectrale de A .

Chapitre 1

Préliminaires et notions fondamentales

Ce chapitre introductif contient certaines notions et résultats bien connus sur le calcul matriciel et quelques propriétés des matrices symétriques (en particulier, semi-définie-positives), ainsi que des notions de base de l'analyse convexe et de la programmation mathématique qui seront utiles par la suite.

1.1 Quelques notions sur les 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 . L'espace vectoriel des matrices carrées symétriques d'ordre n est noté $\mathbb{S}^n = \{A \in M_n(\mathbb{R}) / A^T = A\}$.

1.1.1 Valeurs propres et spectre

1. Pour une matrice $A \in M_n(\mathbb{R})$, le polynôme à variable λ ,

$$p_A(\lambda) = \det(\lambda I_n - A),$$

est appelé polynôme caractéristique de A . Les racines de $p_A(\lambda)$ sont dites les valeurs propres de A .

2. Il est important de signaler que le déterminant d'une matrice A est le produit de ses valeurs

propres.

3. Le spectre de $A \in M_n(\mathbb{R})$, est l'ensemble des valeurs propres de A ,

$$\sigma(A) = \{\lambda \in \mathbb{R}, (\lambda I_n - A) \text{ est singulière (i.e., n'est pas inversible)}\}.$$

4. Le rayon spectral de A est la plus grande valeur propre de A en valeur absolue

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

5. Pour nos besoins, il convient d'écrire les valeurs propres dans l'ordre croissant

$$\lambda_{\min}(A) = \lambda_1(A) \leq \lambda_2(A) \leq \dots \leq \lambda_n(A) = \lambda_{\max}(A),$$

et pour A et B deux matrices de \mathbb{S}^n , on a :

(a) $\lambda_{\min}(A + B) \geq \lambda_{\min}(A) + \lambda_{\min}(B)$.

(b) $\lambda_{\max}(A + B) \leq \lambda_{\max}(A) + \lambda_{\max}(B)$.

6. Soit la matrice $A \in M_n(\mathbb{R})$, alors on a :

(a) Si A est orthogonale, alors les valeurs propres de A sont -1 ou 1 .

(b) A est régulière (ou inversible) si et seulement si 0 n'est pas une valeur propre de A .

7. Soit A une matrice inversible de $M_n(\mathbb{R})$ alors :

(a) λ est une valeur propre de A si et seulement si λ est une valeur propre de A^T .

(b) λ est une valeur propre de A si et seulement si λ^{-1} est une valeur propre de A^{-1} .

(c) λ est une valeur propre de A si et seulement si $\lambda + \beta$ est une valeur propre de $A + \beta I_n$.

(d) λ est une valeur propre de A si et seulement si $\lambda\beta$ est une valeur propre de βA .

Proposition 1.1 *Toutes les valeurs propres notées $\lambda_i(A)$ d'une matrice symétrique $A \in \mathbb{S}^n$ sont réelles et de plus il existe une matrice orthogonale $P \in M_n(\mathbb{R})$ (dite de passage) 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 Trace d'une matrice

Définition 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 .

Il est important de signaler que la trace est une fonction linéaire, de plus elle vérifie les propriétés suivantes :

1. $\forall A, B \in M_n(\mathbb{R}), \text{Tr}(AB) = \text{Tr}(BA)$.
2. $\forall A \in M_n(\mathbb{R}), \text{Tr}(A) = \text{Tr}(A^T)$.
3. $\forall A, B \in M_n(\mathbb{R}) : A \sim B \implies \text{Tr}(A) = \text{Tr}(B)$.

1.1.3 Produit scalaire et normes

On commence par la définition du produit scalaire de deux vecteurs.

Définition 1.2 Étant donné deux vecteurs $x, y \in \mathbb{R}^n$, le produit scalaire euclidien est noté et défini par

$$\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i.$$

De même, on définit un produit scalaire sur l'ensemble des matrices carrées réelles.

Définition 1.3 Soient $A, B \in M_n(\mathbb{R})$, le produit scalaire de A et B noté $A \bullet B$ est défini par :

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

Énonçons maintenant la notion de norme vectorielle.

Définition 1.4 La norme vectorielle est une application de \mathbb{R}^n dans \mathbb{R}_+ , notée par $\|\cdot\|$ et vérifie les conditions suivantes :

1. $\forall x \in \mathbb{R}^n : \|x\| = 0 \iff x = 0$.
2. $\forall \lambda \in \mathbb{R}, \forall x \in \mathbb{R}^n : \|\lambda x\| = |\lambda| \|x\|$.
3. $\forall x, y \in \mathbb{R}^n : \|x + y\| \leq \|x\| + \|y\|$.

La norme matricielle associée au produit scalaire de deux matrices est définie par :

Définition 1.5 Soient $A, B \in M_n(\mathbb{R})$, l'application $\|\cdot\| : \mathbb{R}^{n \times n} \longrightarrow \mathbb{R}_+$ est appelée norme matricielle si elle vérifie les conditions suivantes :

1. $\|A\| = 0 \iff A = 0, \forall A \in M_n(\mathbb{R})$.
2. $\|\lambda A\| = |\lambda| \|A\|, \forall \lambda \in \mathbb{R}$.
3. $\|A + B\| \leq \|A\| + \|B\|, \forall A, B \in M_n(\mathbb{R})$.
4. $\|AB\| \leq \|A\| \|B\|, \forall A, B \in M_n(\mathbb{R})$.

les normes matricielles usuelles pour toute $A \in M_n(\mathbb{R})$ sont :

$$\|A\|_1 = \max_{j=1}^n \left(\sum_{i=1}^n |a_{ij}| \right), \|A\|_\infty = \max_{i=1}^n \left(\sum_{j=1}^n |a_{ij}| \right), \text{ et } \|A\|_2 = \sqrt{\max_{i=1}^n |\lambda_i|} = \sqrt{\rho(A^T A)},$$

la dernière norme est appelée la norme euclidienne, où $\lambda_i, i = 1, \dots, n$, sont les valeurs propres de la matrice $A^T A$.

On utilisera également la norme de Frobenius :

$$\|A\|_F = \sqrt{A \bullet A} = \sqrt{\sum_{i=1}^n \sum_{j=1}^n (a_{ij})^2}, \forall A \in M_n(\mathbb{R}).$$

On note que si : $A \in \mathbb{S}^n$, alors on obtient facilement les résultats suivants :

$$\|A\|_F = \sqrt{(A^T A)} = \sqrt{\text{Tr}(A^2)} = \sqrt{\sum_{i=1}^n \lambda_i^2(A)},$$

et

$$\|A\|_2 = \sqrt{\rho(A^2) |\lambda_i|} = \max_{i=1}^n |\lambda_i(A)|,$$

de plus,

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n} \|A\|_2,$$

et pour toute norme matricielle on a :

$$\rho(A) \leq \|A\|.$$

1.1.4 Matrices semi-définies positives

Enonçons maintenant la notion d'une matrice semi-définie positive. Puis, on présente certaines propriétés qui seront utilisées par la suite.

Définition 1.6 Soit $A \in \mathbb{S}^n$,

- ▷ A est dite semi-définie positive, et on écrira $A \succeq 0$, si et seulement si : $x^T A x \geq 0, \forall x \in \mathbb{R}^n$, et on note par \mathbb{S}_+^n l'ensemble des matrices semi-définies positives.
- ▷ A est dite définie positive, et on écrira $A \succ 0$ si et seulement si : $x^T A x > 0, \forall x \in \mathbb{R}^n - \{0\}$, et on note par \mathbb{S}_{++}^n l'ensemble des matrices définies positives.

Théorème 1.1 Soit $A \in \mathbb{S}^n$, alors les propositions suivantes sont équivalentes :

1. $A \in \mathbb{S}_+^n$ (resp. $A \in \mathbb{S}_{++}^n$).
2. $\lambda_i(A) \geq 0$ (resp. $\lambda_i(A) > 0$), $\forall i = 1, \dots, n$.
3. $\exists B \in M_n(\mathbb{R})$ telle que : $A = B^T B$.
4. Pour une suite arbitraire $A_i \in \mathbb{S}^i, i = 1, \dots, n$, 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 1.2 Soit $A \in \mathbb{S}_{++}^n$, alors il existe une matrice unique $B \in \mathbb{S}_{++}^n$, telle que : $A = B^2$, et on la note souvent par $B = A^{\frac{1}{2}}$. De plus, B est appelée la racine carrée de A .

Proposition 1.3 Soient $A, B \in \mathbb{S}_+^n$, alors :

1. $A \succeq B \iff A - B \succeq 0$.
2. $A + B \succeq B$.
3. $A^{\frac{1}{2}} B A^{\frac{1}{2}} \succeq 0$.

4. $Tr(AB) \leq Tr(A)Tr(B)$.

5. $Tr(AB) \geq 0$.

Propriétés

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

2. Pour tout $A \in \mathbb{S}_+^n$, il existe $i \in \{1, \dots, n\}$ tel que $a_{ii} = \max_{i,j \in \{1, \dots, n\}} |a_{ij}|$.

3. Si $A \in \mathbb{S}_+^n$ et $a_{ii} = 0$ pour un certain $i \in \{1, \dots, n\}$, alors $a_{ij} = 0$ pour tout $j \in \{1, \dots, n\}$.

4. Soit $B \in M_n(\mathbb{R})$ une matrice inversible, alors : $A \in \mathbb{S}_+^n$ (resp. \mathbb{S}_{++}^n) $\iff B^T A B \in \mathbb{S}_+^n$ (resp. \mathbb{S}_{++}^n).

5. On a l'équivalence suivante : $A \succeq 0 \iff A \bullet B \geq 0, \forall B \succeq 0$.

6. On a aussi : $A \succ 0 \iff A \bullet B > 0, \forall B \succeq 0$ non nulle.

7. Si $A, B \in \mathbb{S}_+^n$, alors : $A \bullet B = 0 \iff AB = 0 \iff \frac{1}{2}(AB + BA) = 0$.

8. Soit $A, B \in \mathbb{S}^n$, alors :

(a) Si $A \succeq 0$, alors : $\|A\|_F \leq Tr(A)$, et $n(\det(A))^{\frac{1}{n}} \leq Tr(A)$.

(b) Si $C, D \in \mathbb{S}^n$ telle que $C - A \succeq 0$ et $D - B \succeq 0$, alors : $Tr(AB) \leq Tr(CD)$.

(c) $A \succeq B \iff C^T A C \geq C^T B C, \forall C \in M_n(\mathbb{R})$.

(d) Si $A \succeq I_n$ alors : A est inversible et $I_n \succeq A^{-1}$.

(e) Si $B \succeq A \succ 0$ alors B est inversible et $A^{-1} \succeq B^{-1}$.

9. $\lambda_{\min}(A) \lambda_{\max}(B) \leq \lambda_{\min}(A) tr(B) \leq A \bullet B \leq n \lambda_{\max}(A) tr(B) \leq n^2 \lambda_{\max}(A) \lambda_{\max}(B)$.

Théorème 1.2 Si $A \in \mathbb{S}_{++}^p, C \in \mathbb{S}^n, B \in \mathbb{R}^{p \times n}$ alors

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succeq 0 \iff C - B^T A^{-1} B \succeq 0.$$

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

La matrice $S = C - B^T A^{-1} B$ s'appelle complément de Schur de A .

En particulier, pour tout $x \in \mathbb{R}^n$ et $X \in \mathbb{S}^n$

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \iff X - xx^T \succeq 0$$

1.2 Rappel d'analyse convexe

La notion de convexité est un outil mathématique important pour l'étude théorique et numérique des problèmes d'optimisation. On présente dans cette section quelques notions d'analyse convexe d'usage courant.

1.2.1 Ensembles affines

Définition 1.7 Un sous ensemble F de \mathbb{R}^n est dit affine si :

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

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 1.8 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 1.3 Toute partie affine de \mathbb{R}^n contient ses combinaisons affines :

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

1.2.2 Fonctions affines

Définition 1.9 Une fonction f est dite affine sur $F \subset \mathbb{R}^n$ si :

$$\forall x, y \in \mathbb{R}^n, \forall \lambda \in \mathbb{R} : f[(1 - \lambda)x + \lambda y] = (1 - \lambda)f(x) + \lambda f(y),$$

Théorème 1.4 Toute fonction affine f de \mathbb{R}^n dans \mathbb{R}^m est de la forme $f(x) = Ax + b$; où $A \in M_{m,n}(\mathbb{R})$ et $b \in \mathbb{R}^m$.

Proposition 1.4 Une fonction affine f de \mathbb{R}^n dans \mathbb{R}^m est linéaire si et seulement si $f(0) = 0$, i.e., $b = 0$.

1.2.3 Ensembles convexes

Définition 1.10 Un sous ensemble C de \mathbb{R}^n est dit convexe si :

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

Autrement dit, si le segment de droite joignant deux points quelconques $x, y \in C$, c-à-d : $[x, y] = \{(1 - \lambda)x + \lambda y, 0 \leq \lambda \leq 1\}$ est entièrement inclus dans C .

Définition 1.11 C est un polyèdre convexe s'il est de la forme :

$$C = \{x \in \mathbb{R}^n : a_i^t x \leq b_i, i = 1, \dots, m\},$$

où a_i est un vecteur non nul de \mathbb{R}^n et b_i un scalaire pour $i = 1, \dots, m$.

Remarque 1.1 C peut s'écrire sous la forme matricielle suivante :

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

où A est une matrice de $\mathbb{R}^{m \times n}$ et b un vecteur de \mathbb{R}^m .

1.2.4 Fonctions convexes

Définition 1.12 La fonction $f : \mathcal{C} \rightarrow \mathbb{R}$ est dite convexe équivaut à dire que

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y), \quad \forall \lambda \in [0, 1], \quad \forall x, y \in \mathcal{C}.$$

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

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

et si l'inégalité au dessus est stricte, alors f est dite strictement convexe $\forall x \neq y$:

1.2.5 Fonction mid-convexe

Définition 1.13 Soit f une fonction réelle de la variable réelle définie sur un intervalle I non réduit à un point. On dit que « la fonction f est mid-convexe sur \mathcal{C} » si on a :

$$\forall (x, y) \in \mathcal{C}^2, \quad f\left(\frac{x+y}{2}\right) \leq \frac{f(x) + f(y)}{2}.$$

► Soit f une fonction réelle de la variable réelle définie sur \mathcal{C} non réduit à un point. Si f est continue et mid-convexe sur \mathcal{C} alors f est convexe sur I .

1.2.6 Caractérisation d'une fonction convexe différentiable

Si $f \in C^1(\mathcal{C})$, où \mathcal{C} est un ensemble convexe alors on a les équivalences suivantes :

1. f est convexe si et seulement si

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in \mathcal{C}.$$

2. f est convexe si et seulement si

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0, \quad \forall x, y \in \mathcal{C}.$$

3. De plus, f est dite strictement convexe si l'une ou l'autre des inégalités précédentes sont strictes pour $x \neq y$.
4. f est convexe si et seulement si $\nabla^2 f(x)$ (le Hessien de f) est semi-défini positif sur \mathcal{C} (c'est à dire que $y^t \nabla^2 f(x) y \geq 0, \forall x, y \in \mathcal{C}$ ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont positives).
5. f est strictement convexe si et seulement si $\nabla^2 f(x)$ est défini positif sur \mathcal{C} (c'est à dire que $y^t \nabla^2 f(x) y > 0, \forall x, y \in \mathcal{C}$ et $y \neq 0$ ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives).

1.2.7 Fonction barrière

Définition 1.14 Soit $D \subset \mathbb{R}^n$. On appelle fonction barrière associée à D toute fonction B_r définie sur $\text{int}(D)$ telle que :

1. B_r est continue.
2. $B_r(x) \geq 0, \forall x \in \text{int}(D)$.
3. $\lim B_r(x) = +\infty$, quand $x \rightarrow x^* \in \text{Fr}(D)$.

Les fonctions barrières les plus utilisées sont les fonctions logarithmiques définies par :

$$B_r(x) = -\sum_{i=1}^n \ln(h_i(x))$$

Exemple 1.1 Pour

$$D = \{x \in \mathbb{R}^n : h_i(x) \leq 0, i = 1, \dots, m\},$$

avec $h_i \in C^1(\mathbb{R}^n)$ et

$$\text{int}(D) = \{x \in \mathbb{R}^n : h_i(x) < 0, i = 1, \dots, m\}.$$

On peut prendre

$$B_{r1}(x) = -\sum_{i=1}^m \ln \frac{1}{h_i(x)},$$

et

$$B_{r2}(x) = -\sum_{i=1}^m \log(-h_i(x)).$$

Remarque 1.2 Dans la programmation mathématique, la fonction barrière classique la plus utilisée est la fonction barrière logarithmique.

1.2.8 Cônes

Définition 1.15 On dit que K est un cône si :

$$\lambda K \subset K, \forall \lambda > 0.$$

Un cône K est dit :

1. Pointé si $0 \in K$.
2. Saillant si $K \cap (-K) \subset \{0\}$.
3. Solide si $\text{Int}(K) \neq \emptyset$.
4. Convexe si $K + K \subset K$.

Proposition 1.5 L'ensemble \mathbb{S}_+^n est un cône convexe fermé saillant et solide. Et l'ensemble \mathbb{S}_{++}^n est un cône convexe.

Comme exemples de cône, on peut citer les orthants positifs \mathbb{R}_+^n , \mathbb{R}_{++}^n l'ensemble des polynômes positifs, l'ensemble des fonctions réelles qui sont positives sur une partie donnée de leur ensemble de définition, les ensembles de matrices symétriques définies positives, semi-définies positives.

Proposition 1.6 \triangleright L'intersection quelconque de cônes convexes est un cône convexe.

\triangleright Le produit cartésien de cônes convexes est un cône convexe.

1.3 Programmation mathématique

On donne, dans cette partie, les outils de base de l'optimisation. On rappelle quelques définitions élémentaires et les théorèmes fondamentaux d'existence et d'unicité de minima. Des conditions d'optimalité sont aussi présentées.

1.3.1 Définitions

Problème d'optimisation

Sous sa forme générale, un problème d'optimisation s'écrit comme suit :

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

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continue, $\emptyset \neq C \subseteq \mathbb{R}^n$ est l'ensemble des contraintes, et si $C = \mathbb{R}^n$, ce problème est appelé problème d'optimisation sans contraintes.

Programme mathématique

Un programme mathématique est un problème d'optimisation qui consiste à trouver une solution du problème qui maximise ou minimise une fonction donnée sous un ensemble des contraintes. En général, un programme mathématique est défini par :

$$(PM) \quad \begin{cases} \min_x f(x), \\ x \in \mathcal{F}, \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continue est appelée fonction objective de (PM) et $\mathcal{F} \subseteq \mathbb{R}^n$ est l'ensemble des contraintes (ou des solutions admissibles), dit aussi domaine de faisabilité. Souvent \mathcal{F} est présenté comme suit :

$$\mathcal{F} = \{x \in \mathbb{R}^n, h_j(x) \leq 0, j = 1, 2, \dots, p \text{ et } g_i(x) = 0, i = 1, 2, \dots, n\},$$

avec g_i, h_j sont des fonctions de $\mathbb{R}^n \rightarrow \mathbb{R}$.

Solution réalisable (faisable) on appelle solution réalisable de (PM) tout point x^0 vérifiant les contraintes (i.e., $x^0 \in \mathcal{F}$).

Solution optimale globale : Une solution réalisable qui optimise l'objectif sur \mathcal{F} est dite solution optimale globale de (PM) (i.e., $x^* \in \mathcal{F}$ est un minimum global de f si et seulement si pour tout $x \in \mathcal{F}$, $f(x^*) \leq f(x)$). L'ensemble des solutions optimales globales est noté :

$\arg \min_{\mathcal{F}} f(x)$, (L'argument du minimum).

Solution optimale locale Un point $x^* \in \mathcal{F}$ est une solution optimale locale de (PM) si :

$$\exists \vartheta \text{ (voisinage) de } x^* \text{ telque : } f(x^*) \leq f(x), \forall x \in \vartheta.$$

Contrainte saturée (active) Une contrainte d'inégalité $h_j(x) \leq 0, \forall j$, est dite saturée (active) en $x^* \in \mathcal{F}$ si $h_j(x^*) = 0$. Une contrainte d'égalité est par définition saturée en tout point x de \mathcal{F} .

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.

- (PM) est un problème différentiable si les fonctions f, g_i, h_j sont toutes différentiables.
- (PM) est un problème convexe si f et h_j sont convexes et g_i affines.

Qualification des contraintes

La Qualification des contraintes est satisfaite pour tout $x^* \in \mathcal{F}$ dans l'un des trois cas suivants :

- Les contraintes sont affines (\mathcal{F} est un polyèdre convexe).
- La condition de (**Slater**) : Si \mathcal{F} est convexe (c-à-d h_j sont convexes et g_i sont affines) et $\text{int}(\mathcal{F}) \neq \emptyset$, c-à-d :

$$\exists x^0 : h_j(x^0) < 0, \text{ et } g_i(x^0) = 0, \forall i, j.$$

- La condition de (**Magazarian-Fromowitz**) : Si les gradients des contraintes saturées en $x^* \in \mathcal{F}$ sont linéairements indépendants, alors les contraintes sont qualifiées en x^* .

Théorème 1.5 *Pour un programme convexe, tout optimum local est un optimum global.*

La convexité et la différentiabilité jouent un rôle très important dans la programmation mathématique.

Définition 1.16 Soit une contrainte inégalité $h_j(x) \leq 0$. Si x^* satisfait $h_j(x^*) < 0$, on dit que la contrainte est inactive en x^* . Si x^* satisfait $h_j(x^*) = 0$, on dit que la contrainte est active ou saturée en x^* .

1.3.3 Résolution d'un programme mathématique (PM)

Principaux résultats d'existence et d'unicité

Théorème 1.6 (Weierstrass) Si $\mathcal{F} \subseteq \mathbb{R}^n$ est compact (fermé et borné), et f est continue sur \mathcal{F} , alors (PM) admet au moins une solution optimale globale $x^* \in \mathcal{F}$.

Corollaire 1.1 Si \mathcal{F} est non vide et fermé, et f est continue et coercive sur \mathcal{F} (au sens que $f(x) \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$), alors (PM) admet une solution optimale globale.

Corollaire 1.2 Si f est une fonction inf-compacte, alors (PM) admet une solution optimale globale.

Théorème 1.7 Si \mathcal{F} est convexe et f est strictement convexe, alors il existe au plus une solution optimale de (PM).

Remarque 1.3 La stricte convexité n'assure pas l'existence de la solution mais assure l'unicité.

Conditions d'optimalités

Considérons le problème (PM) suivant :

$$(PM) \quad \begin{cases} \min_x f(x), \\ h_j(x) \leq 0, j = 1, \dots, p, \\ g_i(x) = 0, i = 1, \dots, n, \\ x \in \mathbb{R}^n, \end{cases}$$

on associe à ce problème la fonction lagrangienne suivante :

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{j=1}^p \lambda_j h_j(x) + \sum_{i=1}^n \mu_i g_i(x), x \in \mathbb{R}^n,$$

où : λ_j et μ_i sont des réels dits multiplicateurs de Lagrange tel que : $\lambda_j \in \mathbb{R}^+, j = 1, \dots, p$ et $\mu_i \in \mathbb{R}, i = 1, \dots, n$.

La théorie de Karush-Kuhn-Tucker (**K.K.T**) permet d'écrire les conditions nécessaires d'optimalité pour tout problème d'optimisation avec contraintes possédant une fonction objectif différentiable.

Théorème 1.8 (Karush-KuhnTucker) Si x^* est une solution optimale locale de (PM) satisfaisant l'une des conditions de qualifications précédentes, alors il existe des multiplicateurs $\lambda_j \in \mathbb{R}^+, j = 1, \dots, p$ et $\mu_i \in \mathbb{R}, i = 1, \dots, n$. tels que :

$$(K.K.T) \quad \begin{cases} \nabla f(x^*) + \sum_{j=1}^p \lambda_j \nabla h_j(x^*) + \sum_{i=1}^n \mu_i \nabla g_i(x^*) = 0, & (\text{condition d'optimalité}), \\ \lambda_j h_j(x^*) = 0, \quad j = 1, \dots, p, & (\text{condition de complémentarité}), \\ g_i(x^*) = 0, \quad i = 1, \dots, n. \end{cases}$$

Remarque 1.4 1) Si les contraintes ne sont pas qualifiées en x^* , les conditions de (K.K.T) ne s'appliquent pas (x^* peut être optimal sans vérifier ces conditions).

2) Si (PM) est convexe, les conditions de (K.K.T) sont à la fois nécessaires et suffisantes pour que x soit un minimum global.

1.3.4 Dualité lagrangienne

Le dual de (PM) est le programme mathématique (DM) suivant :

$$(DM) \quad \begin{cases} \sup_{\lambda, \mu} \inf_{x \in \mathcal{F}} \mathcal{L}(x, \lambda, \mu), \\ \nabla_x \mathcal{L}(x, \lambda, \mu) = 0, \\ \lambda \in \mathbb{R}^n, \mu \in \mathbb{R}^p. \end{cases}$$

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

1.3.5 Méthode de Newton pour résoudre un système non linéaire

La résolution du (PM) revient à résoudre le système d'équation non linéaire suivant :

$$\begin{cases} \nabla f(x) + \sum_{j=1}^p \lambda_j \nabla h_j(x) + \sum_{i=1}^n \mu_i \nabla g_i(x), \\ \lambda_j h_j(x), & j = 1, \dots, p, \\ g_i(x), & i = 1, \dots, n. \end{cases}$$

Posons

$$F(x, \lambda, \mu) = \begin{pmatrix} \nabla f(x) + \sum_{j=1}^p \lambda_j \nabla h_j(x) + \sum_{i=1}^n \mu_i \nabla g_i(x) = 0, \\ \lambda_j h_j(x) = 0, & j = 1, \dots, p, \\ g_i(x) = 0, & i = 1, \dots, n. \end{pmatrix}$$

où

$$\begin{aligned} F : \mathbb{R}^{2n+p} &\longrightarrow \mathbb{R}^{2n+p} \\ (x, \lambda, \mu) &\longmapsto \left(\nabla f(x) + \sum_{j=1}^p \lambda_j \nabla h_j(x) + \sum_{i=1}^n \mu_i \nabla g_i(x), \lambda_j h_j(x), g_i(x) \right) \end{aligned}$$

Les méthodes les plus populaires appliquées pour la résolution d'un système non linéaire est la méthode de Newton. Dans ce qui suit décrivons son principe.

Soit $g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ une fonction continue, différentiable et soit $J(x)$ la matrice jacobienne de la fonction g . Alors nous considérons le système non linéaire suivant :

$$g(x) = 0.$$

A partir d'un vecteur x^0 de \mathbb{R}^n et l'utilisation de la formule suivante :

$$x^{k+1} = x^k - J(x^k)^{-1} g(x^k),$$

on construit une suite de point définie par :

$$x^{k+1} = x^k + d^k,$$

où d^k est le vecteur de direction, solution du système :

$$J(x^k)d^k = -f(x^k).$$

1.3.6 Méthodes de directions admissibles

Cette classe de méthodes résout un problème de minimisation non linéaire en se déplaçant d'un point de l'ensemble des contraintes de (PM) \mathcal{F} vers un autre de ses points au coût inférieur. Elles fonctionnent selon le principe suivant : étant donné un élément x^k de \mathcal{F} , une direction d^k est générée telle que pour un $\alpha^k > 0$ et suffisamment petit, les propriétés suivantes sont assurées :

1. $x^k + \alpha^k d^k$ appartient toujours à \mathcal{F} .
2. $f(x^k + \alpha^k d^k)$ est inférieure à $f(x^k)$.

Une fois d^k déterminée, α^k s'obtient par minimisation monodimensionnelle pour que le déplacement dans la direction d^k soit optimal, mais cette fois-ci il est nécessaire d'imposer une borne supérieure sur la valeur de α^k afin de ne pas sortir de \mathcal{F} . Cela définit le nouveau point x^{k+1} et le processus est recommencé.

1.3.7 Méthode de résolution d'un programme mathématique

On peut classer les méthodes de résolution d'un programme mathématique en trois catégories

1) Méthodes de type gradient

a) Gradient conjugué : Cette méthode a été proposée par Hestenes (1952) pour résoudre un système linéaire à matrice définie-positive, puis généralisée par Fletcher et Reeves (1964) pour résoudre des problèmes d'optimisation non linéaires, elle est connue par son efficacité pour minimiser une fonction quadratique sans contraintes. Dans le cas contraint, un changement de variable simple permet de se ramener au cas sans contraintes, en effet : x^0 un point satisfaisant les contraintes ($Ax^0 = 0$) et posons $x = x^0 + P_A y$ tel que $P_A = I - A^T(AA)^{-1}A$ est l'opérateur de la projection sur le noyau de la matrice A . Le principe de cette méthode est de construire

progressivement des directions d_0, d_1, \dots, d_k mutuellement conjuguées par rapport à la matrice Hessienne $(\nabla^2 f(x))$ de la fonction objectif du problème d'optimisation : $d_i \nabla f(x) d_j = 0, \forall i \neq j, i, j = 0, 1, \dots, k$.

b) Gradient projeté (Rosen 1960) : le principe de cette méthode est de projeter à chaque itération le gradient sur la frontière du domaine réalisable. Il faut signaler que cette méthode est conçue pour un programme plus général de la forme :

$$\begin{cases} \min f(x), \\ Ax = b, \\ x \geq 0. \end{cases}$$

où f est différentiable non nécessairement convexe.

2) Méthodes simpliciales

Parmi les méthodes simpliciales on cite, celle de gradient réduit due à Wolfe. C'est une extension directe de la méthode du simplexe, appliquée à la programmation quadratique. De ce fait, elle présente les mêmes inconvénients à savoir complexité exponentielle.

3) Méthodes de points intérieurs

Conjointement aux méthodes décrites précédemment, il existe actuellement des méthodes de points intérieurs pour la résolution d'un problème d'optimisation convexe. Ce sont des extensions des méthodes développées pour la programmation linéaire (affines, projectives et de trajectoire centrale). Les problèmes d'initialisation, le coût de l'itération et le choix de la direction de descente deviennent plus pesants.

On distingue trois classes fondamentales de méthodes de points intérieurs à savoir :

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

b) Méthodes de réduction du potentiel : La fonction potentiel joue un grand rôle dans le développement des méthodes de points intérieurs. L'algorithme de Karmarkar appliqué au programme linéaire sous forme standard utilise une fonction potentiel de la forme $(n + 1) \log (c^T x - z) - \sum_{i=1}^n \log (x_i)$, où z est une borne inférieure de la valeur optimale de l'objectif. Karmarkar prouve la convergence et la polynômialité de son algorithme et a montré que cette fonction potentiel est réduite à chaque itération par au moins une constante. Depuis 1987, les chercheurs introduisent des fonctions du potentiel de type primales-duales, parmi lesquelles, celle de Tanabe, Todd et Ye [48] définie par $\Phi_p(x, s) = p \log (x^T s) - \sum_{i=1}^n \log (x_i s)$, pour $p > n$. Cette fonction a joué un rôle très important dans le développement des algorithmes de réduction du potentiel après 1988. Les algorithmes correspondants à ces méthodes possèdent une complexité polynômiale, nécessitant $O(\sqrt{n} |\log \varepsilon|)$ itérations pour réduire le saut de dualité.

c) Méthodes de la trajectoire centrale : Elles ont été introduites à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles possèdent de bonnes propriétés théoriques :

une complexité polynômiale et une convergence superlinéaire.

Les algorithmes de la trajectoire centrale restreignent les itérés à un voisinage du chemin central, ce dernier est un arc de points strictement réalisables.

Plusieurs chercheurs essaient de généraliser le principe de ces méthodes pour la programmation non linéaire. En 1989, Megiddo [40] a proposé un algorithme primal-dual de trajectoire centrale pour la programmation linéaire avec une généralisation pour le problème de complémentarité linéaire (*LCP*). Kojima et all. [28] ont développé un algorithme primal-dual pour la programmation linéaire, une extension pour le (*LCP*) est proposée par les mêmes chercheurs en 1989 avec la complexité $O\left(\sqrt{n} \log \frac{1}{\varepsilon}\right)$ itérations.

Chapitre 2

Programmation semi-définie

Dans ce chapitre, on s'intéresse à l'étude des problèmes de minimisation de fonctions linéaires sous l'ensemble des contraintes linéaires dont l'inconnu X est une matrice carrée symétrique semi-définie-positif ($X \in \mathbb{S}_+^n$).

2.1 Formulation du problème

Un programme semi-défini est un programme mathématique avec :

- Une fonction objectif linéaire.
- Un ensemble des variables x_{ij} .
- Un ensemble des contraintes linéaires.
- Une contrainte de semi-défini positivité.

Définition 2.1 *Le programme semi-défini sous forme standard s'écrit :*

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

où C, X et A_i ($i = 1, \dots, m$) sont des matrices dans \mathbb{S}^n , et $b = (b_1, \dots, b_m) \in \mathbb{R}^m$.

Définition 2.2 Une matrice $X \in \mathbb{S}^n$ est dite réalisable (resp. strictement réalisable) pour (SDP) si :

$$A_i \bullet X = b_i, \quad i = 1, \dots, m \text{ et } X \succeq 0, \text{ (resp. } A_i \bullet X = b_i, \quad i = 1, \dots, m \text{ et } X \succ 0).$$

Et on note par $\mathcal{F}_p = \{X \in \mathbb{S}^n, X \text{ réalisable}\}$, (resp. $\mathcal{F}_p^+ = \{X \in \mathbb{S}^n, X \text{ strictement réalisable}\}$), l'ensembles des solutions réalisables primales, (resp. l'ensembles des solutions strictement réalisables primales) pour (SDP).

Définition 2.3 La valeur optimale primale de (SDP) est définie par :

$$p^* = \inf [C \bullet X : X \in \mathbb{S}_+^n, A_i \bullet X = b_i, i = 1, \dots, m]$$

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

$$X^* \in \mathcal{F}_p \text{ et } C \bullet X^* = p^*.$$

2.2 Cas particuliers

Plusieurs programmes mathématiques peuvent se formuler en un problème (SDP). Nous présentons ici quelques uns.

2.2.1 Programmation linéaire

En pratique, personne ne veut transformer un programme linéaire en un programme (SDP) pour la résolution numérique, mais du point de vue théorique, un programme linéaire peut être considéré comme un cas particulier de problème (SDP). En effet :

Un programme linéaire (PL) est un problème sous la forme :

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

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

Pour transformer le programme linéaire (PL) en un programme (SDP), on définit :

$$X = \text{diag}(x), C = \text{diag}(c) \text{ et } A_i = \text{diag}(A_{i,\cdot}) \forall i \in \{1, \dots, m\},$$

où $A_{i,\cdot}$ désigne la $i^{\text{ème}}$ ligne de la matrice A . Nous réécrivons alors la fonction objectif linéaire sous la forme :

$$c^T x = C \bullet X,$$

et pour les contraintes, on a :

$$Ax = b \iff A_i \bullet X = b_i, \quad i = 1, \dots, m.$$

De même

$$x_i \succeq 0, i = 1, \dots, n \iff x \in \mathbb{R}_+^n \iff X \in \mathbb{S}_+^n \iff X \succeq 0.$$

Ainsi, (PL) en variable x peut être réécrit sous la forme standard d'un (SDP) en variable matricielle X comme suit :

$$(SDP) \quad \begin{cases} \min C \bullet X, \\ A_i \bullet X = b_i, \quad i = 1, \dots, m, \\ X \succeq 0, \end{cases}$$

2.2.2 Programmation quadratique

Les problèmes d'optimisation quadratiques peuvent aussi se formuler en programme (SDP) .

Définition 2.4 Un programme quadratique convexe (PQC) sous contraintes quadratiques convexes est défini par :

$$(PQC) \quad \begin{cases} \min x^T Q_0 x + b_0^T x + c_0, \\ x^T Q_i x + b_i^T x + c_i \leq 0, \quad i = 1, \dots, m, \\ x \in \mathbb{R}^n, \end{cases}$$

où $Q_i \in \mathbb{S}_+^n; b_i \in \mathbb{R}^n, c_i \in \mathbb{R}, i = 1, \dots, m.$

Pour transformer (PQC) en (SDP), on procède de la manière suivante :

Pour la fonction objectif on rajoutant une variable auxiliaire réelle y de sorte que

$$x^T Q_0 x + b_0^T x + c_0 \leq y, y \in \mathbb{R}.$$

Donc la résolution de (PQC) revient à la résolution du problème

$$(PQC) \quad \begin{cases} \min y, \\ x^T Q_0 x + b_0^T x + c_0 \leq y, \\ x^T Q_i x + b_i^T x + c_i \leq 0, \quad i = 1, \dots, m, \\ x \in \mathbb{R}^n, \end{cases}$$

On transforme ensuite les contraintes en s'appuyant sur le théorème de Schur. En effet,

Comme $Q_i \in \mathbb{S}_+^n$, elle possède une unique racine carrée $M_i \in \mathbb{S}_+^n$ d'où $Q_i = M_i^2$, et on a :

$$x^T Q_0 x + b_0^T x + c_0 \leq y \iff \begin{pmatrix} I & M_0 x \\ x^T M_0^T & c_0 - b_0^T x + y \end{pmatrix} \succeq 0.$$

$$x^T Q_i x + b_i^T x + c_i \leq 0 \iff \begin{pmatrix} I & M_i x \\ x^T M_i^T & c_i - b_i^T x \end{pmatrix} \succeq 0.$$

Le problème (PQC) se transforme alors comme suit

$$\begin{cases} \min y \\ \begin{pmatrix} I & M_0 x \\ x^T M_0^T & c_0 - b_0^T x + y \end{pmatrix} \succeq 0 \\ \begin{pmatrix} I & M_i x \\ x^T M_i^T & c_i - b_i^T x \end{pmatrix} \succeq 0 \\ x \in \mathbb{R}^n, y \in \mathbb{R}, \end{cases}$$

où il y a $(n + 1)$ variables inconnues (x, y) et $(m + 1)$ contraintes d'inégalité matricielles linéaires.

2.3 Domaines d'application

La programmation semi-définie a trouvé de nombreuses applications en ingénierie, statistiques, graphes,... etc. Quelques unes de ces applications sont présentées dans la partie suivante.

2.3.1 Optimisation des valeurs propres

Recherche des valeurs propres extrêmes

Considérons A une matrice symétrique réelle, le problème de recherche de la valeur propre minimale d'une matrice symétrique peut se formuler comme suit :

$$\begin{aligned}\lambda_{\min}(A) &= \min_{\|x\|=1} x^T A x \\ &= \min_{x \in \mathbb{R}^m, x \neq 0} \frac{x^T A x}{\|x\|^2}.\end{aligned}$$

Posant $X = x x^T$, alors

$$Tr(X) = \sum_{i=1}^n x_i^2 = \|x\|^2 = 1.$$

Le problème devient alors

$$(SDP) \quad \begin{cases} \min A \bullet X = \lambda_{\min}(A), \\ Tr(X) = 1, \\ X \succeq 0. \end{cases}$$

De même, le problème de la plus grande valeur propre peut s'écrire comme :

$$\begin{aligned}\lambda_{\max}(A) &= \max_{\|x\|=1} x^T A x \\ &= \max_{x \in \mathbb{R}^m, x \neq 0} \frac{x^T A x}{\|x\|^2},\end{aligned}$$

et sa formulation en (SDP) est :

$$\begin{cases} \max A \bullet X = \lambda_{\max}(A), \\ \text{Tr}(X) = 1, \\ X \succeq 0. \end{cases}$$

Problèmes de Min-Max des valeurs propres

Le problème suivant a été étudié dans [13, 5] comme suit :

$$\lambda^* = \min \lambda_{\max}(C + A(x)).$$

Où : $C \in M_n$, et

$$\begin{aligned} A : \mathbb{R}^n &\longrightarrow M_n \\ x &\longmapsto A(x) = \sum_{i=1}^n x_i A_i. \end{aligned}$$

On a la propriété $A \preceq \lambda I_n \iff \lambda_{\max}(A) \leq \lambda$, tel que $A \in \mathbb{S}^n$ et $\lambda \in \mathbb{R}$.

Le problème min-max s'écrit sous forme d'un problème (*SDP*) comme suit :

$$\begin{cases} \min \lambda, \\ \lambda_{\max}(C + A(x)) \leq \lambda, \\ x \in \mathbb{R}^n. \end{cases} \iff \begin{cases} \min \lambda, \\ \lambda I - C - A(x) \succeq 0, \\ x \in \mathbb{R}^n. \end{cases}$$

Norme spectrale d'une matrice

Soit $A \in M_n$. On considère sa norme spectrale

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

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

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

On utilise ici le théorème du complément de Schur.

Minimisation de la somme des r plus grandes valeurs propres

Pour minimiser la somme des r plus grandes valeurs propres de la matrice symétrique $A(x)$ dépend affinement de x i.e., $A(x) = A_0 + x_1A_1 + \dots + x_kA_k$, avec $A_i \in \mathbb{S}^n$, $i = 0, \dots, k$, on résout le problème de programmation semi-définie suivant

$$\left\{ \begin{array}{l} \min rt + Tr(X), \\ tI_n + X - A(x) \succeq 0, \\ X \succeq 0. \end{array} \right.$$

Où $X \in \mathbb{S}^n$ et $t \in \mathbb{R}$.

Pour plus de détail voir [50].

2.3.2 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 :

$$\begin{bmatrix} I & Ax + b \\ (Ax + b)^T & c^T x + d \end{bmatrix} \succeq 0.$$

Cette dernière inégalité peut être exprimée par :

$$F(x) = F_0 + x_1F_1 + \dots + x_kF_k,$$

avec

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

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

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

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 :

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

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

2.3.3 Problème de programmation non linéaire

Exemple 2.1 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 (SDP) comme suit :

$$(SDP) \begin{cases} \min C \bullet X, \\ A \bullet X = b, \\ X \succeq 0, \end{cases}$$

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

Exemple 2.2 Soit le programme non linéaire suivant :

$$\begin{cases} \min \frac{(c^T x)^2}{d^T x}, \\ Ax + b \geq 0, \end{cases}$$

où on suppose que $d^T x > 0$. Ce problème se reformule alors de la manière suivante :

$$\begin{cases} \min t, \\ Ax + b \geq 0, \\ \frac{(c^T x)^2}{d^T x} \leq t \iff \begin{pmatrix} t & c^T x \\ c^T x & d^T x \end{pmatrix} \succeq 0. \end{cases}$$

On obtient alors le problème (SDP) suivant :

$$\begin{cases} \min t, \\ \begin{pmatrix} \text{diag}(Ax + b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{pmatrix} \succeq 0. \end{cases}$$

Exemple 2.3 Soit le programme non linéaire suivant :

$$\begin{cases} \min h(\alpha) = \frac{10}{\cos \alpha} + 6 - 5 \tan \alpha, \\ \alpha \in]0; \frac{\pi}{2}[. \end{cases}$$

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

$$(SDP) \begin{cases} \min C \bullet X, \\ A \bullet X > 1, \\ X \succeq 0, \end{cases}$$

avec

$$X = \begin{pmatrix} \frac{1}{\cos \alpha} & 0 & 0 \\ 0 & \tan \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 10 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 6 \end{pmatrix} \text{ et } A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

2.4 Dualité en programmation semi-définie

La dualité en (*SDP*) linéaire est très similaire à la dualité en (*PL*) classique à quelques différences près.

Soit le problème (*SDP*) linéaire standard

$$(SDP) \begin{cases} \min C \bullet X, \\ A_i \bullet X = b_i, i = 1, \dots, m, \\ X \succeq 0. \end{cases}$$

Pour obtenir le problème dual de (*SDP*), on considère la fonction lagrangienne :

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

et on a :

$$\max_{y \in \mathbb{R}^m} q(y) = \begin{cases} \max b^T y, & \text{si } C - \sum_{i=1}^m y_i A_i \succeq 0, \\ -\infty, & \text{ailleurs.} \end{cases}$$

D'où par convention le dual du problème (*SDP*) est :

$$(DSDP) \begin{cases} \max b^T y, \\ C - \sum_{i=1}^m y_i A_i \succeq 0, i = 1, \dots, m, \\ y \in \mathbb{R}^m. \end{cases}$$

C'est un programme semi-défini qui s'écrit aussi :

$$(DSDP) \begin{cases} \max b^T y, \\ S + \sum_{i=1}^m y_i A_i = C, i = 1, \dots, m, \\ y \in \mathbb{R}^m, S \succeq 0. \end{cases}$$

Définition 2.5 Une solution réalisable de (DSDP) est le couple $(y, S) \in \mathbb{R}^m \times \mathbb{S}^n$ tel que :

$$S \succeq 0, \sum_{i=1}^m y_i A_i + S = C, i = 1, \dots, m.$$

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

$$S \succ 0, \sum_{i=1}^m y_i A_i + S = C, i = 1, \dots, m.$$

On désigne par $\mathcal{F}_d = \{(y, S) \in \mathbb{R}^m \times \mathbb{S}^n, (y, S) \text{ réalisable}\}$, l'ensemble des solutions réalisables duales de (DSDP). Et $\mathcal{F}_d^+ = \{(y, S) \in \mathbb{R}^m \times \mathbb{S}^n, (y, S) \text{ strictement réalisable}\}$, l'ensemble des solutions strictement réalisables duales de (DSDP).

Définition 2.6 La valeur optimale de (DSDP) est définie par :

$$d^* = \sup_y \left[b^T y : C - \sum_{i=1}^m y_i A_i \in \mathbb{S}_+^n, i = 1, \dots, m, y \in \mathbb{R}^m \right].$$

(y^*, S^*) est une solution optimale duale de (DSDP) si :

$$(y^*, S^*) \in \mathcal{F}_d, \text{ et } b^T y^* = d^*.$$

2.4.1 Relations primales-duales pour la programmation semi-définie

Comme pour la programmation linéaire, le problème (*SDP*) peut s'écrire sous plusieurs formes, le tableau suivant présente les différents types de problèmes duales correspondants

Minimisation	Maximisation
Variables	Contraintes
matrice ou scalaire ≥ 0	matrice ou scalaire \leq
matrice ou scalaire ≤ 0	matrice ou scalaire \geq
matrice $\succeq 0$	matrice \preceq
matrice $\preceq 0$	matrice \succeq
matrice ou scalaire non astreint	matrice ou scalaire =
Contraintes	Variables
matrice ou scalaire \leq	matrice ou scalaire ≥ 0
matrice ou scalaire \geq	matrice ou scalaire ≤ 0
matrice \preceq	matrice $\succeq 0$
matrice \succeq	matrice $\preceq 0$
matrice ou scalaire =	matrice ou scalaire non astreint

2.4.2 Dualité faible

Proposition 2.1 Soit $X \in \mathcal{F}_p$ et $(y, S) \in \mathcal{F}_d$, alors :

$$C \bullet X - b^T y = S \bullet X \geq 0, \text{ et } p^* \geq d^*.$$

Preuve 1) On a

$$\begin{aligned}
 C \bullet X - b^T y &= C \bullet X - \sum_{i=1}^m b_i y_i \\
 &= C \bullet X - \sum_{i=1}^m y_i (A_i \bullet X) \\
 &= \left(C - \sum_{i=1}^m y_i A_i \right) \bullet X \\
 &= S \bullet X \geq 0, \text{ car } S \succeq 0, \text{ et } X \succeq 0,
 \end{aligned}$$

donc $C \bullet X - b^T y \geq 0, \forall X \in \mathcal{F}_p$ et $\forall (y, s) \in \mathcal{F}_d$.

2) On a

$$\begin{aligned}
 C \bullet X \geq b^T y, \quad \forall X \in \mathcal{F}_p \text{ et } \forall y \in \mathcal{F}_d &\implies C \bullet X^* \geq b^T y, \forall (y, s) \in \mathcal{F}_d \\
 &\implies p^* = C \bullet X^* \geq b^T y^* = d^* \\
 &\implies p^* \geq d^*,
 \end{aligned}$$

d'où la dualité faible. ■

Remarque 2.1 $S \bullet X$ toujours positive est appelé l'écart ou "**Saut de dualité**".

Remarque 2.2 Certains résultats de la Programmation Linéaire ne sont pas valables en (SDP).

Exemple 2.4 Considérons le problème semi-défini suivant :

$$(SDP) \begin{cases} \min C \bullet X, \\ A_i \bullet X = b_i, \quad i = 1, 2, \\ X \in \mathbb{S}_+^3. \end{cases}$$

\mathbb{S}_+^3 : l'espace des matrices symétriques 3×3 .

$$\begin{aligned}
 C &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, X = \begin{pmatrix} x_{11} & x_{21} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\
 A_2 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.
 \end{aligned}$$

Ce problème s'écrit aussi sous la forme suivante :

$$(SDP) \left\{ \begin{array}{l} \min (1 - x_{12}), \\ \begin{pmatrix} 0 & x_{21} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 - x_{12} \end{pmatrix} \in \mathbb{S}_+^3. \end{array} \right.$$

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 $p^* = 1$.

Le problème dual est :

$$(DSDP) \left\{ \begin{array}{l} \max b^T y, \\ C - \sum_{i=1}^2 y_i A_i \in \mathbb{S}_+^3, \\ y \in \mathbb{R}^2, \end{array} \right.$$

qui correspond à la forme :

$$(DSDP) \left\{ \begin{array}{l} \max 2y_2, \\ \begin{pmatrix} -y_1 & -y_2 & 0 \\ -y_2 & 0 & 0 \\ 0 & 0 & 1 - 2y_2 \end{pmatrix} \in \mathbb{S}_+^3. \end{array} \right.$$

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 $d^* = 0$.

Les problèmes (SDP) et $(DSDP)$ 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 $(DSDP)$ ne suffit pas pour avoir les mêmes valeurs optimales.

En conséquences, l'existence d'une solution optimale pour l'un des deux problèmes (SDP) ou son dual $(DSDP)$ 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.

2.4.3 Dualité forte

Théorème 2.1 1. Si le problème (SDP) est strictement réalisable, c-à-d : $\exists X \in \mathcal{F}_p^+$ alors $p^* = d^*$. Si en outre p^* est fini, alors l'ensemble des solutions optimales duales de $(DSDP)$ est compact non vide.

2. Si le problème $(DSDP)$ est strictement réalisable, c-à-d : $\exists (y, S) \in \mathcal{F}_d^+$ alors $p^* = d^*$. Si en outre d^* est fini, alors l'ensemble des solutions optimales primales de (SDP) est compact non vide.

2.5 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 $(DSDP)$ sous la forme d'une condition dite de complémentarité.

Théorème 2.2 Soient $X^* \in \mathcal{F}_p$ et $(y^*, S^*) \in \mathcal{F}_d$ de saut de dualité

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

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

Le problème de complémentarité s'écrit comme suit :

$$(PC) \begin{cases} \langle A_i, X \rangle = b_i, & i = 1, \dots, m, & X \succeq 0, \\ S + \sum_{i=1}^m y_i A_i = C, & S \succeq 0, \\ XS = 0. \end{cases}$$

Remarque 2.3 Comme (SDP) est un problème convexe, la solution du problème de complémentarité est un optimum global pour (SDP) .

2.6 Méthodes de résolution de (SDP)

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 [12] et Nesterov et Nemirovskii [50]. Alizadeh [12] 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 [50] ont présenté la théorie profonde des méthodes de points intérieurs qui est basée sur des fonctions barrières auto-concordantes.

2.6.1 Méthodes 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.

Derrière le terme points intérieurs découlent trois différents types de méthodes

1. Les méthodes affines.
2. Les méthodes de réduction du potentiel.
3. Les méthodes de trajectoire centrale.

a) Méthodes affines

L'idée remonte en 1967 par Dikin [19], Puis reprise et développée par plusieurs chercheurs au milieu des années 80. Il s'agit pratiquement de l'algorithme de Karmarkar sans fonction potentiel et sans transformation projective, on utilise une transformation affine et on remplace la contrainte de non négativité par un ellipsoïde qui contient le nouvel itéré. L'algorithme correspondant à cette méthode est d'une structure simple, malheureusement, il n'est pas facile de démontrer la polynomialité. À cet égard, on trouve, en 1967 Dikin [19] à prouvé la convergence de la méthode affine primale sous des hypothèses de non dégénérescence. Actuellement la convergence pour les méthodes affines primales ou duales n'est pas prouvée. Par contre, en 1990 Monteiro et all. [42] ont démontré que la méthode affine primale-duale est de complexité polynomiale ($O(nL^2)$); où L représente le nombre de bits requis pour stocker (et traiter) les données.

b) Méthodes de réduction du potentiel

Ces méthodes sont le fruit direct d'une grande partie des études acharnées menées par plusieurs chercheurs vers la fin des années 80. En 1995, Alizadeh a proposé une méthode primale-duale de réduction du potentiel [12]. En 2003 et inspirant des travaux de Alizadeh, Benterki [8] 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. En effet, supposons à l'itération k , qu'on dispose d'une solution strictement réalisable $X_k \in \mathbb{S}_{++}^n$.

Comme X_k est symétrique et définie positive, il existe une matrice triangulaire inférieure L_k tel que $X_k = L_k L_k^T$.

La transformation projective est définie comme suit :

$$\begin{aligned} T_k : \mathbb{S}^n &\longrightarrow \mathbb{S}^n \times \mathbb{R}^r \\ X &\longmapsto T_k(X) = (\bar{X}, \bar{x}) \end{aligned}$$

où

$$\bar{X} = \frac{(n+r)L_k^{-1}XL_k^{-T}}{r+\langle X_k^{-1}, X \rangle} \quad \bar{x} = \frac{(n+r)}{r+\langle X_k^{-1}, X \rangle} e_r \quad e_r = (1, \dots, 1)^T \in \mathbb{R}^r.$$

Propriétés de T_k :

▷ $T_k(X_k) = (I, e_r)$.

▷ T_k est bijective et on a : $X = T_k^{-1}(\bar{X}, \bar{x}) = \frac{L_k \bar{X} L_k^T}{e_r^T \bar{x}}$.

Appliquons T_k sur le problème (SDP), On obtient le 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

$$\bar{C} = L_k^T C L_k \in \mathbb{S}^n.$$

$$\bar{c}(z) = -\left(\frac{x}{r} e_r\right) \text{ est un vecteur de } \mathbb{R}^r.$$

$$\bar{A}_i = L_k^T A_i L_k \in \mathbb{S}^n.$$

$$\bar{\mathcal{A}} = \mathcal{A}(L_k \otimes L_k) \text{ est une } m \times n^2 \text{ matrice.}$$

$$\bar{A} = \left(-\frac{1}{r} b e_r^T\right) \text{ est une } m \times r \text{ matrice.}$$

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

La solution optimale (\bar{X}^*, \bar{x}^*) 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 $\bar{\bar{A}}$.

D'où $X = T_k^{-1}(\bar{X}^*, \bar{x}^*)$ est une solution réalisable de (SDP) candidate à l'optimalité.

2. Fonction potentielle primale-duale :

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

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

Cette fonction permet de prouver la convergence polynômiale de l'algorithme proposé par Alizadeh [12].

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

Les méthodes de trajectoire centrale primale-duale ont été introduites à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles ont attiré une grande attention de la part des chercheurs dans le monde entier et elles montrent en général un excellent comportement pratique et théorique (une complexité polynomiale et une convergence super linéaire). On trouve en 1996 ; les travaux de Helmberg et all. [5] qui ont proposé

un algorithme primal-dual de trajectoire centrale pour (SDP) : En 1997 ; Monteiro [45] a proposé une méthode primale-duale de trajectoire centrale et a montré la convergence polynomiale de l'algorithme à court et long pas. En 1999 ; Ji et all. [21] ont étudié la convergence de la méthode de trajectoire centrale de type prédicteur-correcteur. Aussi, Monteiro et all. [42] ont proposé une méthode primale-duale de trajectoire centrale et ont montré la convergence polynomiale vers une solution optimale. Ces travaux se poursuivent à ce jour, on trouve ainsi ceux de Halicka et all. en 2002 [32], qui ont proposé une étude sur la convergence de la méthode de trajectoire centrale en optimisation semi-définie. En 2007 Koulaei et all. [34] ont proposé une extension de l'algorithme de Mehrotra prédicteur-correcteur basé sur la direction de Nesterov et todd. En 2012, Liu et all. [7] ont présenté un nouveau algorithme de type correcteur de second ordre pour (SDP) et ont prouvé la convergence polynomiale de ce dernier pour la direction de Nesterov et todd.

Le principe de ces méthodes est de minimiser le saut de dualité en résolvant le système paramétrisé des conditions d'optimalité des problèmes (SDP) et ($DSDP$) suivant :

$$(SDP)_\mu \begin{cases} \langle A_i, X \rangle = b_i, & i = 1, \dots, m, \\ S + \sum_{i=1}^m y_i A_i = C, \\ XS = \mu I, \quad X, S \succeq 0. \end{cases}$$

Le système $(SDP)_\mu$ admet une solution unique notée par $(X(\mu), y(\mu), S(\mu))$, ($\mu > 0$ fixé) sous les hypothèses :

- 1) $\mathcal{F}^+ \neq \emptyset$.
- 2) $A_i, i = 1, \dots, n$ sont linéairement indépendantes.

La trajectoire centrale qui est défini par l'ensemble des solutions $(X(\mu), y(\mu), S(\mu))$, converge vers la solution optimale quand μ tend vers 0.

On utilise la méthode de Newton pour résoudre le système non linéaire $(SDP)_\mu$. L'objectif est d'obtenir des directions primales et duales ΔX , Δy et ΔS , respectivement en résolvant le système linéaire suivant :

$$\begin{cases} \langle A_i, \Delta X \rangle = b_i, & i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta S = 0, \\ X \Delta S + S \Delta X = \mu I - XS, & X, S \succ 0. \end{cases}$$

L'itération de Newton complète est définie par :

$$X^+ = X + \Delta X, \quad y^+ = y + \Delta y, \quad S^+ = S + \Delta S.$$

Et doit être strictement réalisable (i.e., $X^+ \in \mathcal{F}_p^+$ et $(y^+, S^+) \in \mathcal{F}_d^+$).

Mais comme X^+ n'est pas toujours symétrique, plusieurs chercheurs ont proposé des directions symétriques pour remédier ce problème, On cite par exemples les travaux de Zhang [52], Hellemberg et al., Kojima et al. et Monteiro [5, 30, 45], Alizadeh-Heaberly-Overton [13] et Nesterov-Todd [31].

Actuellement, plusieurs chercheurs bai et autres [53, 38, 35, 26] ont introduit la notion des fonctions noyaux pour trouver une classe de directions, à travers laquelle les auteurs ont pu améliorer la complexité algorithmique de méthodes à petit et grand-pas.

Chapitre 3

Algorithme de points intérieurs pour un problème d'optimisation semi défini (*SDO*) basé sur une nouvelle fonction noyau

Les méthodes de points intérieurs *IPMs* démarre avec un point positif arbitraire et la faisabilité est atteint comme l'optimalité est approché. Le choix du point de départ dans *IPM* est cruciale pour la performance. Lustig [20] et Tanabe [25] ont été les premiers à présenter *IPMs* pour l'optimisation linéaire (*LO*). Kojima et al. [29] ont été les premiers qui ont prouvé la convergence globale d'un primal - dual *IPM* pour *LO*. Zhang [51] a été le premier qui a présenté la complexité polynomiale pour l'optimisation linéaire (*LO*). Les méthodes primale-duale de points intérieurs (*IPM*) pour la programmation semi définie (*SDO*) ont été largement étudiés, le lecteur est renvoyé à Klerk [10] pour plus d'eclaircissement. Récemment, une méthode de points intérieurs non réalisable de programmation linéaire (*PL*) a été présenté par Roos [6]. Certaines extensions des méthodes de points intérieurs ont été réalisées par Mansouri et Roos [17]. Mansouri et Roos [18] étendu cet algorithme de programmation semi-définie en utilisant une étape de faisabilité spécifique. La fonction de barrière est déterminée par une fonction simple unidimensionnel, appelé fonction noyau. Bai et all. [54] introduit une nouvelle fonction de barrière

qui n'est pas une fonction de barrière dans le sens habituel du terme.

Dans ce chapitre, nous définissons une nouvelle fonction de proximité pour (SDO) par une nouvelle fonction noyau. En outre, nous formulons un algorithme de points intérieurs primal-dual pour (SDO) en utilisant la fonction de proximité et de donner son analyse de la complexité. De plus, la complexité obtenue par l'algorithme de long pas est $O(qSn^{\frac{Sg+1}{2Sg}} \log \frac{Tr(X^0 S^0)}{\varepsilon})$ et de petit pas est $O(q^2 S^2 \sqrt{n} \log \frac{Tr(X^0 S^0)}{\varepsilon})$.

Les notations utilisées tout au long du chapitre sont les suivantes : \mathbb{R}^n , \mathbb{R}_+^n et \mathbb{R}_{++}^n désignent respectivement l'ensemble des vecteurs à n composantes, l'ensemble de vecteurs positifs ou nuls et l'ensemble de vecteurs strictement positifs. $\mathbb{R}^{n \times n}$ désigne l'ensemble des matrices réelles. $\|\cdot\|_F$ et $\|\cdot\|_2$ désigne la norme de Frobenius et la norme spectrale de matrices, respectivement. \mathbb{S}_n , \mathbb{S}_n^+ et \mathbb{S}_n^{++} désigne respectivement le cône de matrices symétrique, matrices symétrique semi-définie positive et matrices symétrique définie positive de dimension $n \times n$. E représente la matrice identité. $A \succeq B$ (or $A \succ B$) si $A - B$ is positive semidefinite (ou bien définie positive). Nous utilisons le produit de matrice interne $A \bullet B = Tr(A^T B)$. Pour $Q \in \mathbb{S}_n^{++}$, l'expression $Q^{\frac{1}{2}}$ indique sa racine carrée symétrique. pour $V \in \mathbb{S}_n^+$, on note $\lambda_{\min}(V)$ à la valeur propre minimale de V .

3.1 Description de l'algorithme

3.1.1 Méthode de la trajectoire centrale

Nous considérons le problème d'optimisation semi définie (SDO), dont la forme primale est donnée par :

$$\min \{Tr(CX) : Tr(A_i X) = b_i, i = 1, \dots, m, X \succeq 0\}, \quad (SDO)$$

et son dual

$$\max \left\{ b^t y : \sum_{i=1}^m y_i A_i + S = C, S \succeq 0 \right\}, \quad (SDD)$$

où A_i , $i = 1, \dots, m$ et C sont des matrices symétriques de $\mathbb{R}^{n \times n}$, b et $y \in \mathbb{R}^m$.

Tout au long de ce chapitre, nous formulons les hypothèses suivantes :

Hypothèse 1 : les matrices $A_i, i = 1, \dots, m$ sont linéairement indépendantes

Hypothèse 2 : une solution initiale (X^0, y^0, S^0) tels que :

$$\text{Tr}(A_i X^0) = b_i, i = 1, \dots, m, X^0 \succ 0, \sum_{i=1}^m y_i^0 A_i + S^0 = C, S \succ 0.$$

Nous avons le lemme suivant qui est bien connu :

Lemme 3.1 [23] *Les propriétés suivantes sont équivalentes :*

1. $X \succeq 0, S \succeq 0$ et $\text{Tr}(XS) = 0$,
2. $X \succeq 0, S \succeq 0$ et $\|X^{\frac{1}{2}} S^{\frac{1}{2}}\|^2 = 0$,
3. $X \succeq 0, S \succeq 0$ et $XS = 0$.

Il est bien connu que la recherche d'une solution optimale (X^*, y^*, S^*) de (SDO) et (SDD) est équivalent à résoudre le système suivant :

$$\begin{cases} \text{Tr}(A_i X) = b_i, i = 1, \dots, m, X \succeq 0, \\ \sum_{i=1}^m y_i A_i + S = C, S \succeq 0, \\ XS = 0. \end{cases} \quad (3.1)$$

L'idée de base des méthodes de points intérieurs (IPMs) de type primale-duale consiste à remplacer dans le système (3.1) la troisième équation, la condition de complémentarité par l'équation paramétrisée $XS = \mu E$ avec $\mu > 0$, où E est une matrice identité d'ordre n . Ainsi on considère le système suivant

$$\begin{cases} \text{Tr}(A_i X) = b_i, i = 1, \dots, m, X \succeq 0, \\ \sum_{i=1}^m y_i A_i + S = C, S \succeq 0, \\ XS = \mu E. \end{cases} \quad (3.2)$$

Si le problème (SDO) et (SDD) satisfait les conditions de points strictement intérieur (IPCs), alors pour chaque $\mu > 0$ le système (3.2) admet une solution unique $(X(\mu), y(\mu), S(\mu))$ (voir [30, 50, 22, 27]), que l'on appelle μ -centre de (SDO) et (SDD). L'ensemble de μ -centres, $\Lambda = \{(X(\mu), y(\mu), S(\mu)) / \mu > 0\}$, construit la trajectoire centrale de (SDO) et (SDD).

En générale, les méthodes de points intérieurs (*IPMs*) pour (*SDO*) se composent de deux stratégies : Le premier stratégie consiste à déterminer une solution paramétrisé strictement réalisable et vérifie certaine conditions (condition de proximité) et le second consiste à diminuer le paramètre μ à $\mu^+ = (1 - \theta)\mu$, pour $\theta \in (0, 1)$ pour déterminer la qualité de solution.

3.1.2 Direction de descente

Sans perte de généralité de *IPMs*, nous supposons que $(X(\mu), y(\mu), S(\mu))$ est une solution pour $\mu > 0$. Par exemple, en raison des hypothèses citée dans le paragraphe 3.1.1, on peut supposer cela pour $\mu^0 = \frac{Tr(X^0 S^0)}{n}$, avec $X^0 \succ 0$ et $S^0 \succ 0$. Nous avons ensuite diminuer μ à $\mu = (1 - \theta)\mu$, pour $\theta \in (0, 1)$, et on résout le système suivant :

$$\begin{cases} Tr(A_i \Delta X) = 0, i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta S = 0, \\ X \Delta S + \Delta X S = \mu E - X S. \end{cases} \quad (3.3)$$

Nous considérons le schéma de symétrisation qui donne la direction de Nestrov-Toold (*NT*), on définit la matrice :

$$P = X^{\frac{1}{2}} (X^{\frac{1}{2}} S X^{\frac{1}{2}})^{-\frac{1}{2}} X^{\frac{1}{2}} = S^{-\frac{1}{2}} (S^{\frac{1}{2}} X S^{\frac{1}{2}})^{\frac{1}{2}} S^{-\frac{1}{2}}, \quad (3.4)$$

Et définir aussi

$$D = P^{\frac{1}{2}}, V = \frac{1}{\sqrt{\mu}} D^{-1} X D^{-1} = \frac{1}{\sqrt{\mu}} D S D$$

On remarque que les matrices D et V sont semi définie positive et symétrique, similaire à (*SDO*) [17]. On peut conclure que le système (3.1) a une unique solution symétrique, on définir

$$\bar{A}_i := \frac{1}{\sqrt{\mu}} D A_i D, \quad i = 1, \dots, m, \quad D_X := \frac{1}{\sqrt{\mu}} D^{-1} \Delta X D^{-1} \quad \text{et} \quad D_S := \frac{1}{\sqrt{\mu}} D \Delta S D. \quad (3.5)$$

La direction de descente de (*NT*) est obtenu à partir du système suivant :

$$\begin{cases} \text{Tr}(\bar{A}_i D_X) = 0, \quad i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i \bar{A}_i + D_S = 0, \\ D_X + D_S = V^{-1} - V = -\nabla \Psi_l(V). \end{cases} \quad (3.6)$$

Nous pouvons dire que $\text{Tr}(D_X D_S) = 0$, qui provient des première et seconde équations de (3.6) ou à partir de l'orthogonalité de ΔX et ΔS . La fonction noyau classique est définie comme suit :

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

3.1.3 Algorithme générique de points intérieurs pour (SDO)

On appelle $\psi_l(t)$ la fonction noyau de la fonction barrière logarithmique $\Psi_l(V)$. Dans ce chapitre on remplace $\psi_l(t)$ par une nouvelle fonction $\psi(t)$ qui est défini dans la section suivante et supposons que $\tau \geq 1$.

L'algorithme proposé dans ce chapitre se déroule comme suit : supposons que l'on donne un point strictement réalisable (X, y, S) qui est dans un τ -voisinage de μ -centre. Ensuite, nous diminuons μ à $\mu^+ = (1 - \theta)\mu$, pour $\theta \in (0, 1)$ puis on résout le système de Newton (3.6) pour obtenir la direction de descente. La condition de positivité d'une nouvelle itération est assurée avec le bon choix de la taille du pas α qui est défini dans la section suivante. Cette procédure est répétée jusqu'à ce qu'on trouve une nouvelle itération (X^+, y^+, S^+) qui est dans un τ -voisinage de μ^+ -centre. Alors μ est encore réduite par le facteur $(1 - \theta)$. Ce processus est répété jusqu'à ce que μ est suffisamment petit, i.e., $n\mu \leq \varepsilon$.

Les paramètres τ, θ et le pas de déplacement α doit être choisie de telle sorte que l'algorithme est optimisée dans le sens où le nombre d'itérations requises par l'algorithme est le plus petit possible. Le choix de paramètre θ joue un rôle important dans la théorie et la pratique de *IPMs*. Si θ est une constante indépendante de la dimension n du problème, par exemple $\theta = \frac{1}{2}$, nous appelons l'algorithme de grand pas. Si θ dépend de la dimension du problème, telles que $\theta = \frac{1}{\sqrt{n}}$, alors l'algorithme est appelé algorithme de petit pas.

L'algorithme générique de points intérieurs primal-dual pour (SDO) est donnée comme suit :

Primal-Dual IPM pour (SDO)

Début Algorithme

Initialisation

$\varepsilon > 0$ un paramètre de précision

un paramètre θ , $0 < \theta < 1$

un paramètre de seuil τ , $0 < \tau < 1$,

une solution initiale strictement réalisable (X^0, y^0, S^0) et $\mu^0 = \frac{\text{Tr}(X^0 S^0)}{n}$ tels que $\Psi(X^0, S^0, \mu^0) \leq$

τ .

$k = 0$

$X := X^0, S := S^0, \mu := \mu^0,$

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

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

Tant que $(\Psi(V) > \tau)$ **faire**

Résoudre le système (3.6) pour obtenir $(\Delta X, \Delta y, \Delta S)$,

Déterminer le pas de déplacement α

Prendre

$X := X + \alpha\Delta X$

$y := y + \alpha\Delta y$

$S := S + \alpha\Delta S$

Fin Tant que.

Fin Tant que.

Fin Algorithme.

Dans la section suivante, nous définissons une nouvelle fonction noyau et donner ses propriétés qui sont essentielles à notre analyse de la complexité.

3.2 Fonction noyau et ses propriétés

Soit $\psi : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$ est une fonction noyau, si ψ est deux fois dérivable et satisfait les conditions suivantes [55] :

$$\begin{aligned} \psi'(1) &= \psi(1) = 0, \\ \psi''(t) &> 0, \\ \lim_{t \rightarrow 0^+} \psi(t) &= \lim_{t \rightarrow \infty} \psi(t) = 0. \end{aligned} \tag{3.7}$$

Dans cette étude, nous définissons une nouvelle fonction noyau suivante :

$$\psi_S(t) = \frac{(t^2 - 1)}{2} - \frac{\log(t)}{2} - \frac{1}{2S} \sum_{j=1}^S \frac{t^{1-jq} - 1}{1 - jq}, \quad q > 1, S \in \mathbb{N} \setminus \{0\}. \tag{3.8}$$

Alors, on a les dérivées partielles de la fonction ψ :

$$\begin{aligned} \psi'_S(t) &= t - \frac{1}{2t} - \frac{1}{2S} \sum_{j=1}^S t^{-jq}, \\ \psi''_S(t) &= 1 + \frac{1}{2t^2} + \frac{1}{2S} \sum_{j=1}^S jq t^{-jq-1}, \\ \psi'''_S(t) &= -\frac{1}{t^3} - \frac{1}{2S} \sum_{j=1}^S jq(jq + 1)t^{-jq-2}. \end{aligned} \tag{3.9}$$

D'après les conditions (3.7), $\psi_S(t)$ est bien définie.

$$\psi''_S(t) > 1, \text{ pour } t > 0. \tag{3.10}$$

On remplace la fonction $\Psi_l(V)$ de (3.6) par la fonction $\Psi_S(V)$ suivante :

$$D_X + D_S = -\nabla \Psi_S(V) \tag{3.11}$$

où $\Psi_S(V) = \text{Tr}(\psi_S(V)) = \sum_{i=1}^n \psi_S(\lambda_i(V))$, $\psi_S(t)$ définie dans (3.8). Ainsi, la direction de

descente $(\Delta X, \Delta y, \Delta S)$ est obtenue à l'aide de la résolution du système suivant :

$$\begin{cases} Tr(A_i \Delta X) = 0, i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta S = 0, \\ X \Delta S + \Delta X S = -\mu V \nabla \Psi_S(V). \end{cases} \quad (3.12)$$

On a

$$\begin{aligned} D_X = D_S = 0 & \Leftrightarrow \nabla \Psi_S(V) = 0 \\ & \Leftrightarrow V = E \\ & \Leftrightarrow \nabla \Psi_S(V) = 0 \\ & \Leftrightarrow X = X(\mu), S = S(\mu). \end{aligned} \quad (3.13)$$

Nous utilisons $\Psi_S(V)$ comme fonction de proximité pour mesurer la distance entre l'itération en cours et μ -centrale pour $\mu > 0$. On définit aussi la mesure de proximité basé sur la norme, $\delta(V) : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$, comme suit :

$$\delta(X, S, \mu) = \delta(V) = \frac{1}{2} \|\nabla \Psi_S(V)\| = \frac{1}{2} \|D_X + D_S\|. \quad (3.14)$$

Lemme 3.2 Pour $\psi_S(t)$, nous avons ce qui suit.

- (i) $\psi_S(t)$ est exponentielle convexe pour tous $t > 0$,
- (ii) $\psi_S''(t)$ est monotone décroissante pour tous $t > 0$,
- (iii) $t\psi_S''(t) - \psi_S'(t) > 0$, pour tous $t > 0$.

Preuve Pour (i), par le Lemme 2.1.2 dans [50], il suffit de montrer que la fonction $\psi_S(t)$ satisfait $t\psi_S''(t) + \psi_S'(t) \geq 0$, pour $t > 0$. En utilisant (3.9), on a

$$t\psi_S''(t) + \psi_S'(t) = 2t + \frac{1}{2S} \sum_{j=1}^S (jq - 1) t^{-jq} > 0, \text{ pour } t > 0,$$

et par le Lemme 2.1.2 dans [23], on déduit que $\psi_S(t)$ est exponentiellement convexe.

Pour (ii), en utilisant (3.9), on a $\psi_S'''(t) > 0$, d'où le résultat

Pour (iii), en utilisant (3.9), on a

$$t\psi_S''(t) - \psi_S'(t) = \frac{1}{t} + \frac{1}{2S} \sum_{j=1}^S (1+jq)t^{-jq} > 0, \text{ pour } t > 0.$$

D'où le résultat. ■

Lemme 3.3 Pour $\psi_S(t)$, nous avons ce qui suit.

$$\frac{1}{2}(t-1)^2 \leq \psi_S(t) \leq \frac{1}{2} [\psi_S'(t)]^2, \text{ pour } t > 0, \quad (3.15)$$

$$\psi_S(t) \leq \left[\frac{6+q(S+1)}{8} \right] (t-1)^2, t > 1., \text{ pour } t \geq 1. \quad (3.16)$$

Preuve Pour (3.15), en utilisant (3.7) et (3.10), on a

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

et on a aussi,

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

Pour (3.16), en utilisant le théorème de Taylor, nous avons

$$\begin{aligned}
 \psi_S(t) &= \psi_S(1) + \psi'_S(1)(t-1) + \frac{1}{2}\psi''_S(1)(t-1)^2 + \frac{1}{6}\psi'''_S(\xi)(t-1)^3 \\
 &= \frac{1}{2}\psi''_S(1)(t-1)^2 + \frac{1}{6}\psi'''_S(\xi)(t-1)^3 \\
 &\leq \frac{1}{2}\psi''_S(1)(t-1)^2 \\
 &= \frac{1}{2} \left[\frac{6+q(S+1)}{4} \right] (t-1)^2 = \left[\frac{6+q(S+1)}{8} \right] (t-1)^2, \\
 &= \left[\frac{6+q(S+1)}{8} \right] (t-1)^2.
 \end{aligned}$$

D'où le résultat. ■

Maintenant, nous définissons $\gamma : (0, \infty) \rightarrow (1, \infty)$, la fonction inverse de $\psi_S(t)$ pour $t \geq 1$,

et $\rho : (0, \infty) \rightarrow (0, 1)$, la fonction inverse de $-\frac{1}{2}\psi'_S(t)$ pour $t \in (0, 1)$. Alors nous avons le lemme suivant.

Lemme 3.4 Pour $\psi_S(t)$, nous avons ce qui suit.

$$1 + \sqrt{\frac{8s}{6+q(S+1)}} \leq \gamma(s) \leq 1 + \sqrt{2s}, \quad s \geq 0, \quad (3.17)$$

et

$$\rho(s) \geq \left[\frac{1}{4z+2} \right]^{\frac{1}{sq}}, \quad s \geq 0. \quad (3.18)$$

Preuve Pour (3.17), on pose $s = \psi_S(t)$, $t \geq 1$, i.e., $\gamma(s) = t$, $t \geq 1$,

en utilisant (3.15), on a

$$\psi_S(t) \geq \frac{1}{2}(t-1)^2,$$

alors

$$s \geq \frac{1}{2}(t-1)^2, \quad t \geq 1.$$

Ce qui implique que

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

D'après (3.16), nous avons

$$s = \psi_S(t) \leq \left[\frac{6 + q(S+1)}{8} \right] (t-1)^2, t \geq 1,$$

alors

$$t = \gamma(s) \geq 1 + \sqrt{\frac{8s}{6 + q(S+1)}}.$$

Pour (3.18), on pose $z = -\frac{1}{2}\psi'_S(t)$ pour $t \in (0, 1)$. Par la définition de ρ , on a $\rho(z) = t$ et $2z = -\psi'_S(t)$. Alors

$$2z = -t + \frac{1}{2t} + \frac{1}{2S} \sum_{j=1}^S t^{-jq} > -1 + \frac{1}{2S} \sum_{j=1}^S t^{-Sq} = -1 + \frac{1}{2}t^{-Sq},$$

ce qui implique que

$$t = \rho(z) > \left[\frac{1}{4z+2} \right]^{\frac{1}{Sq}}.$$

D'où le résultat. ■

Lemme 3.5 Soit $\gamma : (0, \infty) \rightarrow (1, \infty)$, la fonction inverse de $\psi_S(t)$ pour $t \geq 1$. Alors on a

$$\Psi_S(\beta V) \leq n\psi_S \left(\beta\gamma \left(\frac{\Psi_S(V)}{n} \right) \right), V \in, \beta \geq 1. \quad (3.19)$$

Preuve En utilisant le théorème 3.2 dans [55], nous obtenons le résultat. ■

Lemme 3.6 Soit $0 \leq \theta \leq 1$, $V^+ = \frac{1}{\sqrt{1-\theta}}V$. Si $\Psi_S(V) \leq \tau$, alors on a

$$\Psi_S(V^+) \leq \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)}. \quad (3.20)$$

Preuve Lorsque $\frac{1}{\sqrt{1-\theta}} \geq 1$ et $\gamma \left(\frac{\Psi_S(V)}{n} \right) \geq 1$, on a $\frac{\gamma \left(\frac{\Psi_S(V)}{n} \right)}{\sqrt{1-\theta}} \geq 1$.

En utilisant le Lemme (3.5), avec $\beta = \sqrt{1-\theta}$, (3.15), (3.16) et $\Psi_S(V) \leq \tau$, on a

$$\begin{aligned}
 \Psi_S(V^+) &\leq n\psi_S\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Psi_S(V)}{n}\right)\right) \\
 &\leq \frac{n}{2}\left(\left[\frac{\sigma\left(\frac{\Psi_S(V)}{n}\right)}{\sqrt{1-\theta}}\right]^2 - 1\right) = \frac{n}{2(1-\theta)}\left(\left[\sigma\left(\frac{\Psi_S(V)}{n}\right)\right]^2 - (1-\theta)\right) \\
 &\leq \frac{n}{2(1-\theta)}\left(\left[1 + \sqrt{2\frac{\Psi_S(V)}{n}}\right]^2 - (1-\theta)\right) \\
 &\leq \frac{n}{2(1-\theta)}\left(2\sqrt{\frac{2\tau}{n}} + 2\frac{\tau}{n} + \theta\right) = \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)}.
 \end{aligned}$$

D'où le résultat. ■

On note par

$$(\Psi_S)_0 = L(n, \theta, \tau) = \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)}, \quad (3.21)$$

alors $(\Psi_S)_0$ est une borne supérieure pour $\Psi_S(V)$ pendant le processus de l'algorithme.

3.3 Convergence de l'algorithme

Le but de ce chapitre est de définir une nouvelle fonction noyau et d'obtenir des nouveaux résultats pour la complexité d'un problème (SDO), en utilisant la fonction de proximité défini par la fonction noyau. En utilisant le concept d'une fonction de matrice [44], la définition de la fonction noyau peut être étendue à toute matrice diagonalisable de valeurs propres positives. En particulier, pour une décomposition de la matrice V donnée par

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

Q_V une matrice non singulière, la fonction de matrice $\psi(V)$ est définie par

$$\psi(V) = Q_V^{-1} \text{diag}(\psi(\lambda_1(V)), \psi(\lambda_2(V)), \dots, \psi(\lambda_n(V))) Q_V. \quad (3.22)$$

Dans cette section, nous calculons un pas de déplacement α efficace et la diminution de la fonction de proximité au cours d'une itération interne et améliorer les résultats de la complexité de l'algorithme, pour $\mu > 0$.

3.3.1 Le pas de déplacement

Soit α est le pas de déplacement, le nouvel itéré est donné comme suit

$$X^+ = X + \alpha \Delta X, \quad y^+ = y + \alpha \Delta y \quad \text{et} \quad S^+ = S + \alpha \Delta S.$$

Soit

$$\begin{aligned} X^+ &= X \left(I + \alpha \frac{\Delta X}{X} \right) = X \left(I + \alpha \frac{D_X}{V} \right) = \frac{X}{V} (V + \alpha D_X), \\ S^+ &= S \left(I + \alpha \frac{\Delta S}{S} \right) = S \left(I + \alpha \frac{D_S}{V} \right) = \frac{S}{V} (V + \alpha D_S). \end{aligned}$$

Alors, on a

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

Cela implique que les valeurs propres de V^+ sont les mêmes que celles de

$$\left((V + \alpha D_X)^{\frac{1}{2}} (V + \alpha D_S) (V + \alpha D_X)^{\frac{1}{2}} \right)^{\frac{1}{2}}.$$

Donc

$$\Psi_S(V^+) = \Psi_S \left(\left((V + \alpha D_X)^{\frac{1}{2}} (V + \alpha D_S) (V + \alpha D_X)^{\frac{1}{2}} \right)^{\frac{1}{2}} \right).$$

Lemme 3.7 (Proposition 5.2.6 dans [23]) Soient $V_1, V_2 \in \mathbb{S}_{++}^n$. Alors on a

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

À partir du lemme (3.7), on obtient

$$\begin{aligned}\Psi_S(V^+) &= \Psi_S \left(((V + \alpha D_X)(V + \alpha D_S))^{\frac{1}{2}} \right), \\ &\leq \frac{1}{2} (\Psi_S(V + \alpha D_X) + \Psi_S(V + \alpha D_S)).\end{aligned}$$

Définir, pour $\alpha > 0$

$$f(\alpha) = \Psi_S(V^+) - \Psi_S(V).$$

Par conséquent, on a $f(\alpha) \leq f_1(\alpha)$, où

$$f_1(\alpha) = \frac{1}{2} (\Psi_S(V + \alpha D_X) + \Psi_S(V + \alpha D)) - \Psi_S(V). \quad (3.23)$$

évidemment,

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

Maintenant, nous traitons avec d'autres concepts pertinents aux fonctions matricielles dans la théorie des matrices [43].

Définition 3.1 [43] Une matrice $X(t)$ est une matrice de fonctions si chaque entrée de $X(t)$ est une fonction de t , c'est-à-dire, $X(t) = [X_{ij}(t)]$.

Les notions de continuité, dérivabilité et intégrabilité s'étendent naturellement aux fonctions de matrice d'une valeur, d'un scalaire en interprétant les composants sage. Ainsi, nous pouvons dire que

$$\frac{d}{dt} X(t) = \frac{d}{dt} [X_{ij}(t)] = X'(t).$$

Supposons que la matrice des fonctions de valeur $H(t)$ et $G(t)$ sont différentiables par rapport à t . Alors on a

$$\frac{d}{dt} (Tr(G(t))) = Tr \left(\frac{d}{dt} G(t) \right) = Tr (G'(t)),$$

$$\frac{d}{dt} ((G(t)H(t)) = G'(t)H(t) + (G(t)H'(t)).$$

Pour chaque fonction $\psi(t)$, notons par $\Delta\psi(t)$ la différence divisée de $\psi(t)$:

$$\Delta\psi(t_1, t_2) = \frac{\psi(t_1) - \psi(t_2)}{t_1 - t_2}, \quad \forall t_1 \neq t_2 \in \mathbb{R}^*.$$

Si $t_1 = t_2$, nous écrivons simplement $\Delta\psi(t_1, t_2) = \psi'(t)$.

Nous allons définir que Q_α est la matrice orthogonale de telle sorte que

$$V + \alpha D_X = Q_\alpha^T \text{diag} (\lambda_1(V + \alpha D_X), \lambda_2(V + \alpha D_X), \dots, \lambda_n(V + \alpha D_X)) Q_\alpha$$

et notons aussi par D_i est une matrice identité. Il résulte de [43, 23] que

$$\frac{d}{d\alpha} (\psi(V + \alpha D_X)) = Q_\alpha^T \left(\sum_{j,k=1}^n \Delta\psi(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) D_j \left(Q_\alpha (V + \alpha D_X)' Q_\alpha^T \right) D_k \right) Q_\alpha. \quad (3.24)$$

Maintenant par le choix de D_i , il contient $Tr (D_j (Q_\alpha (V + \alpha D_X)' Q_\alpha^T) D_k) = 0$, pour $j \neq k$.

Il s'ensuit donc que

$$\begin{aligned} \frac{d}{d\alpha} (\psi(V + \alpha D_X)) &= Tr \left(\sum_{i=1}^n \psi'(\lambda_i(V + \alpha D_X)) D_i \left(Q_\alpha (V + \alpha D_X)' Q_\alpha^T \right) D_i \right), \\ &= Tr \left(Q_\alpha^T \left(\sum_{i=1}^n D_i \psi'(\lambda_i(V + \alpha D_X)) D_i \right) Q_\alpha (V + \alpha D_X)' \right), \\ &= Tr \left(\psi'(V + \alpha D_X) D_X \right). \end{aligned}$$

Et

$$\frac{d}{d\alpha} (\psi(V + \alpha D_S)) = Q_\alpha^T \left(\sum_{j,k=1}^n \Delta\psi(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) D_j (Q_\alpha(V + \alpha D_S)' Q_\alpha^T) D_k \right) Q_\alpha. \quad (3.25)$$

Et $Tr(D_j (Q_\alpha(V + \alpha D_S)' Q_\alpha^T) D_k) = 0$, pour $j \neq k$. Il s'ensuit donc que

$$\begin{aligned} \frac{d}{d\alpha} (\psi(V + \alpha D_S)) &= Tr \left(\sum_{i=1}^n \psi'(\lambda_i(V + \alpha D_S)) D_i (Q_\alpha(V + \alpha D_S)' Q_\alpha^T) D_i \right), \\ &= Tr \left(Q_\alpha^T \left(\sum_{i=1}^n D_i \psi'(\lambda_i(V + \alpha D_S)) D_i \right) Q_\alpha(V + \alpha D_S)' \right), \\ &= Tr \left(\psi'(V + \alpha D_S) D_S \right). \end{aligned}$$

Maintenant, nous pouvons écrire

$$f'_1(\alpha) = \frac{1}{2} Tr \left(\psi'_S(V + \alpha D_X) D_X + \psi'_S(V + \alpha D_S) D_S \right).$$

On obtient

$$f'_1(0) = -2\delta^2.$$

En outre,

$$\begin{aligned} \frac{d^2}{d\alpha^2} (Tr(\psi(V + \alpha D_X))) &= Tr \left(\frac{d}{d\alpha} \left(\psi'(V + \alpha D_X) D_X \right) \right), \\ &= Tr \left(Q_\alpha^T \left(\sum_{j,k=1}^n \Delta\psi'(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) D_j (Q_\alpha(V + \alpha D_X)' Q_\alpha^T) D_k \right) Q_\alpha D_X \right), \\ &= Tr \left(\sum_{j,k=1}^n \Delta\psi'(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) D_j (Q_\alpha D_X Q_\alpha^T) D_k (Q_\alpha D_X Q_\alpha^T) \right), \\ &= Tr \left(\sum_{j,k=1}^n \Delta\psi'(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) (Q_\alpha D_X Q_\alpha^T)_{jk}^2 \right), \\ &\leq \max \left\{ \left| \Delta\psi'(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) \right|, j, k = 1, \dots, n \right\} \|D_X\|^2. \end{aligned}$$

Aussi, on a

$$\frac{d^2}{d\alpha^2} (Tr(\psi(V + \alpha D_S))) = Tr \left(\frac{d}{d\alpha} \left(\psi'(V + \alpha D_S) D_S \right) \right),$$

$$\begin{aligned}
&= Tr \left(Q_\alpha^T \left(\sum_{j,k=1}^n \Delta \psi'(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) D_j (Q_\alpha(V + \alpha D_S)' Q_\alpha^T) D_k \right) Q_\alpha D_S \right), \\
&= Tr \left(\sum_{j,k=1}^n \Delta \psi'(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) D_j (Q_\alpha D_S Q_\alpha^T) D_k (Q_\alpha D_S Q_\alpha^T) \right), \\
&= Tr \left(\sum_{j,k=1}^n \Delta \psi'(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) (Q_\alpha D_S Q_\alpha^T)_{jk}^2 \right), \\
&\leq \max \left\{ \left| \Delta \psi'(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) \right|, j, k = 1, \dots, n \right\} \|D_S\|^2.
\end{aligned}$$

On note par

$$\omega_1 = \max \left\{ \left| \Delta \psi'(\lambda_j(V + \alpha D_X), \lambda_k(V + \alpha D_X)) \right|, j, k = 1, \dots, n \right\},$$

et

$$\omega_2 = \max \left\{ \left| \Delta \psi'(\lambda_j(V + \alpha D_S), \lambda_k(V + \alpha D_S)) \right|, j, k = 1, \dots, n \right\}.$$

Par conséquent,

$$\begin{aligned}
f''(\alpha) &= \frac{1}{2} \frac{d^2}{d\alpha^2} Tr(\psi(V + \alpha D_X) + \psi(V + \alpha D_S)), \\
&\leq \frac{1}{2} (\omega_1 \|D_X\|^2 + \omega_2 \|D_S\|^2).
\end{aligned} \tag{3.26}$$

Lemme 3.8 Soit $\delta(V)$ celui défini au (3.14). Alors on a

$$\delta(V) \geq \sqrt{\frac{\Psi_S(V)}{2}}.$$

Preuve En utilisant (3.15), on a

$$\begin{aligned}
\Psi(V) &= Tr(\psi_S(V)) \\
&= \sum_{i=1}^n \psi_S(\lambda_i(V)) \\
&\leq \sum_{i=1}^n \frac{1}{2} \psi'_S(\lambda_i(V))^2 \\
&= \frac{1}{2} \|\nabla \Psi_S(V)\|^2 \\
&= 2\delta(V)^2,
\end{aligned}$$

on obtient

$$\delta(V) \geq \sqrt{\frac{\Psi_S(V)}{2}}.$$

D'où le résultat. ■

Nous supposons que $\tau \geq 1$. En utilisant le Lemme (3.8) et l'hypothèse que $\Psi_S(V) \geq \tau$, on a $\delta(V) \geq \sqrt{\frac{1}{2}}$.

Nous avons le lemme suivant.

Lemme 3.9 *Soit $f_1(\alpha)$ celui défini au (3.23), $\delta(V)$ celui défini au (3.14) et ψ_S est la fonction noyau défini au (3.8). Alors on a*

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

Preuve On a

$$f_1''(\alpha) \leq 2 \max\{\omega_1, \omega_2\} \delta^2.$$

Il suffit de montrer l'inégalité suivante :

$$\max\{\omega_1, \omega_2\} \leq \psi_S''(\lambda_{\min}(V) - 2\alpha\delta).$$

Nous pouvons choisir $j^*, k^* \in \{1, 2, \dots, n\}$ tel que

$$\omega_1 = \left| \frac{\psi_S'(\lambda_{j^*}(V + \alpha D_X)) - \psi_S'(\lambda_{k^*}(V + \alpha D_X))}{\lambda_{j^*}(V + \alpha D_X) - \lambda_{k^*}(V + \alpha D_X)} \right|.$$

En utilisant le théorème de la valeur moyenne, il existe

$$\eta \in [\min(\lambda_{j^*}(V + \alpha D_X), \lambda_{k^*}(V + \alpha D_X)), \max(\lambda_{j^*}(V + \alpha D_X), \lambda_{k^*}(V + \alpha D_X))],$$

tel que

$$\psi_S''(\eta) = \Delta \psi_S'(\lambda_{j^*}(V + \alpha D_X), \lambda_{k^*}(V + \alpha D_X)).$$

Lorsque D_X est une matrice symétriques, à partir de la définition de δ et norme de Frobenius,

nous avons

$$\begin{aligned}\eta &\geq \min \{ \lambda_{j^*}(V + \alpha D_X), \lambda_{k^*}(V + \alpha D_X) \}, \\ &\geq \lambda_{\min}(V) - 2\alpha\delta,\end{aligned}$$

à fin que, ψ_S'' soit monotone décroissante, on obtient

$$\omega_1 \leq \psi_S''(\lambda_{\min}(V) - 2\alpha\delta).$$

Par la même méthode pour obtenir ω_1 , on obtient aussi

$$\omega_2 = \left| \frac{\psi_S'(\lambda_{j^*}(V + \alpha D_S)) - \psi_S'(\lambda_{k^*}(V + \alpha D_S))}{\lambda_{j^*}(V + \alpha D_S) - \lambda_{k^*}(V + \alpha D_S)} \right|.$$

En utilisant le théorème de la valeur moyenne, il existe

$$\eta \in [\min(\lambda_{j^*}(V + \alpha D_S), \lambda_{k^*}(V + \alpha D_S)), \max(\lambda_{j^*}(V + \alpha D_S), \lambda_{k^*}(V + \alpha D_S))],$$

tel que

$$\psi_S''(\eta) = \Delta \psi_S'(\lambda_{j^*}(V + \alpha D_S), \lambda_{k^*}(V + \alpha D_S)).$$

Lorsque D_S est une matrice symétriques, à partir de la définition de δ et norme de Frobenius, nous avons

$$\begin{aligned}\eta &\geq \min \{ \lambda_{j^*}(V + \alpha D_S), \lambda_{k^*}(V + \alpha D_S) \}, \\ &\geq \lambda_{\min}(V) - 2\alpha\delta,\end{aligned}$$

à fin que, ψ_S'' est monotone décroissante, on obtient

$$\omega_2 \leq \psi_S''(\lambda_{\min}(V) - 2\alpha\delta).$$

Par conséquent,

$$\max \{\omega_1, \omega_2\} \leq \psi_S''(\lambda_{\min}(V) - 2\alpha\delta).$$

on obtient

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

D'où le résultat. ■

puisque $f_1(0) = 0$ et $f_1'(0) = -2\delta(V)^2$, on a

$$\begin{aligned} f(\alpha) &\leq f_1(\alpha) := f_1(0) + f_1'(0)\alpha + \int_0^\alpha \int_0^\xi f_1''(\zeta) d\zeta d\xi, \\ &\leq f_2(\alpha) := f_1(0) + f_1'(0)\alpha + 2\delta^2 \int_0^\alpha \int_0^\xi \psi_S''(\lambda_{\min}(V) - 2\zeta\delta) d\zeta d\xi. \end{aligned}$$

On remarque que $f_2(0) = 0$. Alors

$$f_2'(\alpha) = -2\delta^2 + \delta \left(\psi_S'(\lambda_{\min}(V)) - \psi_S'(\lambda_{\min}(V) - 2\alpha\delta) \right).$$

on a $f_2'(0) = -2\delta^2$, qui est la même valeur de $f_1'(0)$ et $f_2''(\alpha) = 2\delta^2 \psi_S''(\lambda_{\min}(V) - 2\alpha\delta)$, ce qui augmente pour $\alpha \in \left[0, \frac{\lambda_{\min}(V)}{2\delta}\right]$. Ainsi, nous pouvons réécrire $f_2(\alpha)$ comme suit :

$$f_2(\alpha) = f_2(0) + f_2'(0)\alpha + 2\delta^2 \int_0^\alpha \int_0^\xi \psi_S''(\lambda_{\min}(V) - 2\zeta\delta) d\zeta d\xi.$$

Maintenant, en utilisant $f_1'(0) = f_2'(0)$ et $f_1''(\alpha) \leq f_2''(\alpha)$, on peut facilement vérifier que

$$f_1'(\alpha) = f_1'(0) + \int_0^\alpha f_1''(\xi) d\xi \leq f_2'(\alpha).$$

Cela donne que

$$f_1'(\alpha) \leq 0, \text{ si } f_2'(\alpha) \leq 0.$$

Pour calculer le pas de déplacement α telle que la mesure de proximité soit décroît quand nous prenons une nouvelle itération pour $\mu > 0$. Nous voulons calculer le pas de déplacement α qui s'assure que $f_2'(\alpha) \leq 0$ tient avec α le plus grand possible. Puisque $f_2''(\alpha) > 0$, c'est-à-dire $f_2'(\alpha)$ est monotone croissante à α , la plus grande valeur possible à α satisfaisant $f_2'(\alpha) \leq 0$ se produit lorsque $f_2'(\alpha) = 0$, c'est-à-dire

$$\psi_S'(\lambda_{\min}(V)) - \psi_S'(\lambda_{\min}(V) - 2\alpha\delta) = 2\delta \quad (3.27)$$

Lorsque $\psi_S''(t)$ est monotone décroissante, la dérivée de la partie gauche de (3.27) par rapport à $\lambda_{\min}(V)$ est

$$\psi_S''(\lambda_{\min}(V)) - \psi_S''(\lambda_{\min}(V) - 2\alpha\delta) \leq 0.$$

Ainsi, le membre gauche de (3.27) diminue à $\lambda_{\min}(V)$. Cela implique que si $\lambda_{\min}(V)$ devient plus petit, alors α devient plus petit avec δ . On note que

$$\begin{aligned} \delta &= \sqrt{\sum_{i=1}^n (\psi_S'(\lambda_i(V)))^2} \\ &\geq |\psi_S'(\lambda_{\min}(V))| \\ &\geq -\psi_S'(\lambda_{\min}(V)), \end{aligned}$$

et l'égalité est vraie si et seulement si $\lambda_{\min}(V)$ est les coordonnées de $(\lambda_1(V), \lambda_2(V), \dots, \lambda_n(V))$ sont différent de 1 et $\lambda_{\min}(V) < 1$, c'est-à-dire, $\psi_S'(\lambda_{\min}(V)) < 0$.

Par conséquent, la pire situation pour la plus grande valeur de α se produit lorsque $\lambda_{\min}(V)$ satisfait

$$-\psi_S'(\lambda_{\min}(V)) = \delta \quad (3.28)$$

Pour notre propos, nous devons faire face à la pire des cas et si nous supposons que (3.28) est vérifiée. Cela implique

$$\lambda_{\min}(V) = \rho(\delta) \quad (3.29)$$

En utilisant (3.27) et (3.28) on obtient immédiatement

$$-\psi'_S(\lambda_{\min}(V) - 2\alpha\delta) = 4\delta.$$

Par la définition de ρ et l'utilisation de (3.29), la valeur de pas maximal est donnée comme suit :

$$\alpha^* = \frac{\rho(\delta) - \rho(2\delta)}{2\delta}. \quad (3.30)$$

Lemme 3.10 *On considère la définition de ρ et α^* celui défini au (3.30), alors on a*

$$\alpha^* \geq \frac{2S}{2S + S(4\delta + 2)^{\frac{2}{Sq}} + q \sum_{j=1}^S j (4\delta + 2)^{\frac{jq+1}{Sq}}}.$$

Preuve En utilisant le Lemme 4.4 in [55], la définition de $\psi''_S(t)$ et (3.18), on a

$$\begin{aligned} \alpha^* &\geq \frac{1}{\psi''_S(\rho(2\delta))} \\ &= \frac{1}{1 + \frac{1}{2}\rho(2\delta)^{-2} + \frac{q}{2S} \sum_{j=1}^S j (\rho(2\delta))^{-jq-1}} \\ &\geq \frac{1}{1 + \frac{1}{2}(4\delta + 2)^{\frac{2}{Sq}} + \frac{q}{2S} \sum_{j=1}^S j (4\delta + 2)^{\frac{jq+1}{Sq}}} \\ &= \frac{2S}{2S + S(4\delta + 2)^{\frac{2}{Sq}} + q \sum_{j=1}^S j (4\delta + 2)^{\frac{jq+1}{Sq}}} \end{aligned}$$

D'où le résultat. ■

Pour utiliser $\bar{\alpha}$ comme un pas de déplacement par défaut dans l'algorithme, nous définissons $\bar{\alpha}$ comme suit :

$$\bar{\alpha} = \frac{2S}{2S + S(4\delta + 2)^{\frac{2}{Sq}} + q \sum_{j=1}^S j (4\delta + 2)^{\frac{jq+1}{Sq}}}. \quad (3.31)$$

3.3.2 Diminution de la fonction de proximité

Maintenant, on montre que la fonction de proximité Ψ_S avec le pas de déplacement $\bar{\alpha}$ est diminuée. Alors on a le résultat suivant :

Lemme 3.11 [23] Soit $h(t)$ une fonction deux fois différentiable convexe, avec $h(0) = 0$, $h'(0) < 0$ et soit $h(t)$ atteint son minimum (globale) au $t > 0$. Si $h''(t)$ augmente pour $t \in [0, t^*]$, alors

$$h(t) = \frac{th'(0)}{2}.$$

Soit la fonction h tel que

$$h(0) = f_1(0) = 0, \quad h'(0) = f_1'(0) = -2\delta^2, \quad h''(\alpha) = 2\delta^2 \psi_S''(\lambda_{\min}(V) - 2\alpha\delta).$$

Lorsque $f_2(\alpha)$ maintient la condition du lemme ci-dessus, , alors :

$$f(\alpha) \leq f_1(\alpha) \leq f_2(\alpha) \leq \frac{f_2'(0)}{2}\alpha, \quad \text{pour } 0 \leq \alpha \leq \alpha^*.$$

Nous pouvons obtenir la limite supérieure de la valeur décroissante de la proximité dans l'itération interne par le lemme ci-dessus

Théorème 3.1 Soit $\bar{\alpha}$ le pas de déplacement définie dans (3.31) et $\delta = \Psi_S(V) \geq \tau = 1$. Alors, on a

$$f(\bar{\alpha}) \leq -\frac{2S}{8\sqrt{2}(S+8)(1+4qS)} [\Psi_S(v)]^{\frac{Sq-1}{2Sq}}.$$

Preuve Pour tous $\bar{\alpha} \leq \alpha^*$, on a

$$\begin{aligned} f(\bar{\alpha}) &\leq -\bar{\alpha}\delta^2 \\ &= -\frac{(2S)\delta^2}{2S+S(4\delta+2)\frac{2}{S^q}+q\sum_{j=1}^S j(4\delta+2)\frac{j^{q+1}}{S^q}} \\ &\leq -\frac{(2S)\delta^2}{2S(2\delta)+S(2(2\delta)+2(2\delta))\frac{2}{S^q}+q\sum_{j=1}^S j(2(2\delta)+2(2\delta))\frac{j^{q+1}}{S^q}} \\ &\leq -\frac{(2S)\delta^2}{2S(2\delta)+4\frac{2}{S^q}(2\delta)\frac{2}{S^q}+q\sum_{j=1}^S j4\frac{j^{q+1}}{S^q}(2\delta)\frac{j^{q+1}}{S^q}} \\ &\leq -\frac{(2S)\delta^2}{2S(2\delta)\frac{Sq+1}{S^q}+4\frac{2}{S^q}(2\delta)\frac{Sq+1}{S^q}+q\sum_{j=1}^S j4\frac{j^{q+1}}{S^q}(2\delta)\frac{Sq+1}{S^q}} \\ &\leq -\frac{(2S)\delta^2}{2\frac{2S+1}{S^q}\left(2S+4^2+q\sum_{j=1}^S j4\frac{j^{q+1}}{S^q}\right)\delta\frac{Sq+1}{S^q}} \\ &\leq -\frac{(2S)\delta^2-\frac{Sq+1}{S^q}}{2\frac{Sq+1}{S^q}\left(2S+4^2+q\sum_{j=1}^S j4\frac{Sq+1}{S^q}\right)} \\ &\leq -\frac{(2S)\delta\frac{Sq-1}{S^q}}{2^2\left(2S+4^2+q\sum_{j=1}^S j4^2\right)} \end{aligned}$$

$$\begin{aligned}
&= -\frac{(2S)\delta^{\frac{Sq-1}{Sq}}}{2^2(2S+16+16q\sum_{j=1}^S j)} \\
&= -\frac{(2S)\delta^{\frac{Sq-1}{Sq}}}{4(2S+16+8qS(1+S))} \\
&\leq -\frac{(2S)\delta^{\frac{Sq-1}{Sq}}}{4(2(S+8)+8qS(1+S))} \\
&\leq -\frac{2S}{4(2(S+8)+8qS(S+8))} \frac{1}{2^{\frac{Sq-1}{2Sq}}} [\Psi_S(v)]^{\frac{Sq-1}{2Sq}} \\
&\leq -\frac{2S}{8\sqrt{2}(S+8)(1+4qS)} [\Psi_S(v)]^{\frac{Sq-1}{2Sq}} .
\end{aligned}$$

D'où le résultat. ■

3.3.3 Les bornes d'itération

Nous avons besoin de compter le nombre d'itérations internes nécessaires pour revenir à la situation où $\Psi_S(V) \leq \tau$ après une mise à jour de μ . On note $(\Psi_S)_0$ la valeur de $\Psi_S(V)$ après une mise à jour de μ ainsi les valeurs suivantes de la même itération extérieure notée $(\Psi_S)_k$, $k = 1, \dots$. Si K désigne le nombre total d'itérations internes dans l'itération extérieure, puis nous avons

$$(\Psi_S)_0 \leq L = O(n, \theta, \tau), \quad (\Psi_S)_{K-1} > \tau, \quad 0 \leq (\Psi_S) \leq \tau.$$

et selon la (3.19),

$$(\Psi_S)_{k+1} \leq (\Psi_S)_k - \frac{2S}{8\sqrt{2}(S+8)(1+4qS)} (\Psi_S)_k^{\frac{m}{2(m+1)}} .$$

Alors, nous invoquons le lemme suivant du Lemme 14 en[23]

Lemme 3.12 [23] Soient t_0, t_1, \dots, t_k une sequence de nombres positives telle que

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

Où $\beta > 0$, $0 < \nu \leq 1$, alors

$$K \leq \frac{t_0^\nu}{\beta\nu}.$$

Par consequent

$$t_k = (\Psi_S)_k, \quad \beta = \frac{2S}{8\sqrt{2}(S+8)(1+4qS)} \quad \text{et} \quad \nu = 1 - \frac{Sq-1}{2Sq} = \frac{Sq+1}{2Sq},$$

Lemme 3.13 *Soit K le nombre total d'itérations internes dans l'itération extérieure. Alors, on*

a

$$K \leq \frac{8\sqrt{2}q(S+8)(1+4qS)}{1+Sq} (\Psi_S)_0^{\frac{Sq+1}{2Sq}}.$$

Preuve En utilisant le Lemme (3.12), on obtient

$$K \leq \frac{(\Psi_S)_0^v}{\beta\nu} = \frac{8\sqrt{2}q(S+8)(1+4qS)}{1+Sq} (\Psi_S)_0^{\frac{Sq+1}{2Sq}}.$$

D'où le résultat. ■

Maintenant, nous estimons le nombre total d'itérations de notre algorithme.

Théorème 3.2 *Si $\tau \geq 1$, le nombre total d'itérations est d'au plus*

$$\frac{8\sqrt{2}q(S+8)(1+4qS)}{1+Sq} (\Psi_S)_0^{\frac{Sq+1}{2Sq}} \frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}.$$

Preuve Dans l'algorithme, $n\mu \leq \varepsilon$, $\mu^k = (1-\theta)^k \mu^0$ et $\mu^0 = \frac{\text{Tr}(X^0 S^0)}{n}$. Par simple calcul, nous avons

$$K \leq \frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}.$$

Par conséquent, le nombre d'itérations extérieures est majorée par $\frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}$.

multipliant le nombre d'itérations extérieures par le nombre d'itérations internes, nous obtenons une borne supérieure pour le nombre total d'itérations, à savoir,

$$\frac{8\sqrt{2}q(S+8)(1+4qS)}{1+Sq} (\Psi_S)_0^{\frac{Sq+1}{2Sq}} \frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}.$$

Pour l'algorithme de grand pas, on obtient $O(qSn^{\frac{Sq+1}{2Sq}} \log \frac{\text{Tr}(X^0 S^0)}{\varepsilon})$ itérations et $O(q^2 S^2 \sqrt{n} \log \frac{\text{Tr}(X^0 S^0)}{\varepsilon})$ pour petit pas.

D'où le résultat. ■

3.4 Tests numériques

L'algorithme a été testé sur quelques problèmes de référence issus de la bibliothèque de problèmes de test SDPLIB [58]. Ici nous avons utilisé **Nbr** qui signifie le nombre d'itérations produites par l'algorithme. L'implémentation est manipulée en *C++*. Notre tolérance est $\epsilon = 10^{-8}$. Pour le paramètre de mise à jour, nous avons varier $0 < \theta < 1$.

Examples	Size (m,n)	Nbr. of iterations from [58]	Results of our Algorithm				
control1	(21,15)	106	θ	0.15	0.35	0.75	0.95
			Nbr	104	97	46	34
hinf1	(13, 14)	27	θ	0.15	0.35	0.75	0.95
			Nbr	31	27	20	18
hinf2	(13, 16)	43	θ	0.15	0.35	0.75	0.95
			Nbr	39	32	27	23
hinf3	(13, 16)	109	θ	0.15	0.35	0.75	0.95
			Nbr	103	99	78	69
hinf4	(13, 16)	39	θ	0.15	0.35	0.75	0.95
			Nbr	31	29	17	11
hinf5	(13; 16)	42	θ	0.15	0.35	0.75	0.95
			Nbr	37	30	19	15
hinf7	(13; 16)	38	θ	0.15	0.35	0.75	0.95
			Nbr	31	27	23	17
hinf9	(13, 16)	28	θ	0.15	0.35	0.75	0.95
			Nbr	27	19	13	11
hinf10	(21, 18)	57	θ	0.15	0.35	0.75	0.95
			Nbr	49	36	29	17
truss1	(6, 13)	17	θ	0.15	0.35	0.75	0.95
			Nbr	11	6	5	3
truss4	(12, 19)	21	θ	0.15	0.35	0.75	0.95
			Nbr	17	11	9	6

Conclusion

Les méthodes de points intérieurs sont connues par leur efficacité, rapidité de convergence, simplicité algorithmique et capacité de résoudre des problèmes de grandes tailles. L'inconvénient principal dans ce type de méthodes est l'initialisation, c'est-à-dire la détermination d'un point initial qui se trouve à l'intérieur du domaine. La plupart des résultats théoriques supposent que ce point initial est connu, mais numériquement l'obtention de ce point initial prend beaucoup de temps.

Dans cette thèse, nous avons apporté des contributions d'ordre algorithmique, théorique et numérique. En effet, l'introduction du poids et également l'approche non réalisable constituent un remède appréciable pour le problème d'initialisation au niveau des méthodes de trajectoire centrale. Et d'autre part, en utilisant la fonction noyau d'objectif d'atteint une nouvelle classe de direction de descente, condition de proximité et la complexité théorique.

Les résultats obtenus sont très encourageants et donnent lieu à d'autres perspectives dans le domaine de l'optimisation numérique. Les recherches futures pourraient se concentrer sur l'extension de l'optimisation cône symétrique. Enfin, les tests numériques pour (*SDO*) est un travail intéressant pour étudier le comportement d' algorithmme afin d'être comparée avec d'autres approches existantes.

Bibliographie

- [1] A. V. Fiacco ET G. P. McCormick, Nonlinear programming : Sequential unconstrained minimization techniques, John Wiley and Sons, Inc., New York London-Sydney, 1968.
- [2] A. M. Nunez and R. M. Freund, Condition-measure bounds on the behavior of the central trajectory of semidefinite program. *SIAM J. Optim.* 12 ; pp. 818–836, 2001.
- [3] A. Guemmaz, El A. Djeflal, and B. Bounibane, A primal-dual IPMS for SDO problem based on a new kernel function with a logarithmic barrier term, *AMSJ.* 11, pp. 453-471, 2022.
- [4] B. Craven and B. Mond, Linear programming with matrix variables, *Linear Algebra Appl.* 38 ; pp. 73–80, 1981.
- [5] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz, An interior-point method for semidefinite programming, *SIAM J. Optim.* 6 (1996) 342–361.
- [6] C. Roos, A full-Newton step $O(n)$ infeasible interior-point algorithm for linear optimization, *SIAM J. Optim.* 16(4) ; pp. 1110-1136, 2006.
- [7] C. Liu, H. Liu, A new second-order corrector interior-point algorithm for semidefinite programming, *Math Meth Oper Res.* 75 ; pp. 165–183,2012.
- [8] D. Benterki, Résolution des problème de programmation semi-définie par des méthodes de réduction du potentiel, thèse de Doctorat d'état, département de Mathématiques, Université Ferhat Abbas, Sétif-1, Algérie, 2004.

- [9] D. Zhao, M.W. Zhang. A primal-dual large-update interior point algorithm for semidefinite optimization based on a new parametric kernel function. *Statistics, optimization and information computing.* (1); pp. 41-61, 2013.
- [10] E. Klerk, *Aspects of Semidefinite Programming : Interior Point Methods and Selected Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [11] El A. Djefal, *Etude de quelques algorithmes de points interieurs pour la programmation convexe*, Thèse de doctorat, Département de Mathématique, Université Batna 2, 2013.
- [12] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, *SIAM J. Optim.* 5; pp. 13-51, 1995.
- [13] 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; pp. 746–768, 1998.
- [14] G.Q. Wang, Y.Q. Bai. Primal-dual interior-point algorithms for convex quadratic semidefinite optimization. *Nonlinear analysis : theory, methods & applications.* 71 (7-8); pp. 3389-3402, 2009.
- [15] G.Q. Wang, Y.Q. Bai. A new primal-dual path-following interior-point algorithm for semidefinite optimization. *Journal of Mathematical Analysis and Applications.* 353 (1); pp. 339-349, 2009.
- [16] G.Q. Wang, D.T. Zhu. A unified kernel function approach to primal-dual interior point algorithms for convex quadratic SDO. *Numerical Algorithms.* 57 (4); pp. 537-558, 2011.
- [17] H. Mansouri, C. Roos, Simplified $O(nL)$ infeasible interior-point algorithm for linear optimization using full Newton steps, *Optim. Methods Softw.* 22(3); pp. 519-530, 2007.
- [18] H. Mansouri, C. Roos, A new full-Newton step $O(n)$ infeasible interior-point algorithm for semidefinite optimization, *Numer. Algorithms.* 52(2), pp. 225-255, 2009.
- [19] I. I. Dikin, Iterative solution of problems of linear and quadratic programming, *Dokl. Akad. Nauk.* 174; pp. 747–748, 1967.
- [20] I. J. Lustig, Feasible issues in a primal–dual interior point method for linear programming, *Math. Program.* 49; pp. 145-162, 1991.

- [21] J. Ji, F. A. Potra and R. Sheng, On the local convergence of a predictor-corrector method for semidefinite programming, *SIAM J. Optim.* 10; pp. 195–210, 1999.
- [22] J. Sturm, Theory and algorithms of semidefinite programming, in : H. Frenk, C. Roos, T. Terlaky, S. Zhang (Eds.), *High Performance Optimization*, pp. 1-194, Kluwer Academic Publishers, Boston, MA, 1999.
- [23] J. Peng, C. Roos, T. Terlaky, *Self-Regularity. A New Paradigm for Primal–Dual Interior-Point Algorithms*, Princeton University Press, 2002.
- [24] J. Peng, C. Roos, T. Terlaky. Self-regular functions and new search directions for linear and semidefinite optimization. *Mathematical Programming.* 93; pp. 129-171, 2002.
- [25] K. Tanabe, Centered Newton method for linear programming : interior and ‘exterior’ point method, in : K. Tone (Ed.), *New Methods for Linear Programming*, vol. 3, The Institute of Statistical Mathematics, Tokyo, Japan ; pp. 98-100, 1990.
- [26] L. Vandenberghe, S. Boyd, Applications of semidefinite programming, *Applied Numerical Mathematics*, Volume 29, Issue 3; pp. 283-299, 1999.
- [27] L. Tuncel, Potential reduction and primal dual methods, in : H. Wolkowicz, R. Saigal, L. Vandenberghe (Eds.), *Theory, Algorithms and Applications*, in : *Handbook of Semidefinite Programming*, pp. 235-265, Kluwer Academic Publishers, Boston, MA, 2000.
- [28] M. Kojima, S. Mizuno, A. Yoshise, A primal-dual interior point algorithm for linear programming, in : N. Megiddo (Ed.), *Progress in Mathematical Programming : Interior Point and Related Methods*, Springer Verlag, New York, pp. 29–47, 1989.
- [29] M. Kojima, N. Megiddo, S. Mizuno, A primal–dual infeasible-interior-point algorithm for linear programming, *Math. Program.* 61; pp. 263-280, 1993.
- [30] M. Kojima, S. Shindoh, and S. Hara, Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices, *SIAM J. Optim.* 7; pp. 86–125, 1997.
- [31] M. J. Todd, K. C. Toh, R. H. Tütüncü, On the Nesterov-Todd direction in semidefinite programming, *SIAM J. Optim.* 8; pp. 769-796, 1998.
- [32] 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.

- [33] M. El Ghami. New primal-dual interior-point methods based on kernel functions. Phd thesis. Delft University, Netherland, 2005.
- [34] M. H. Koulaei, T. Terlaky, On the extension of a Mehrotra-type algorithm for semidefinite optimization, Technical Report 2007/4, Advanced optimization Lab. Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada.
- [35] 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 ; pp. 99 –115, 2008.
- [36] M. El Ghami, Z.A. Guennoun, S. Bouali, T. Steihaug, , Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term. *J. Comput. Appl. Math.* 236 ; pp. 3613–3623, 2012.
- [37] M.W. Zhang. A large-update interior-point algorithm for convex quadratic semidefinite optimization based on a new kernel function. *Acta Mathematica Sinica. English Series.* 28 (11); pp. 2313-2328, 2012.
- [38] M. Bouafia, D. Benterki, A. Yassine, An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a trigonometric barrier term, *J. Optim. Theory Appl.* 170 ; pp. 528- 545, 2016.
- [39] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica.* 4 ; pp. 373–395, 1984.
- [40] N. Megiddo, Pathways to the optimal set in linear programming, in : N. Megiddo (Ed.), *Progress in Mathematical Programming : Interior Point and Related Methods*, Springer Verlag, New York. 313 ; pp. 158,1989.
- [41] 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.
- [42] R. D. C. Monteiro, I. Adler, M. G. C. Resende, A polynomial time primal dual affine scaling algorithm for linear and convex quadratic programming and its power series extension, *Math. Oper. Res.* 15 ; pp. 191–214, 1990.
- [43] R. A. Horn, C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.

- [44] R. Bellman, Introduction to Matrix Analysis, in : Classics in Applied Mathematics, vol. 12, SIAM, Philadelphia, 1995.
- [45] R. D. C. Monteiro, Primal-dual path-following algorithms for semidefinite programming, SIAM J. Optim. 7; pp. 663–678, 1997.
- [46] R. D. C. Monteiro, Polynomial convergence of primal-dual algorithms for semide-finite programming based on Monteiro and Zhang family of directions, SIAM J. Optim. 8; pp. 797–812, 1998.
- [47] R. D. C. Monteiro and T. Tsuchiya, Polynomial convergence of a new family of primal-dual algorithms for semidefinite programming, SIAM J. Optim. 9; pp. 551–577, 1999.
- [48] S.J. Wright, Primal-dual interior point methodes, SIAM, Philadelphia, USA,1997.
- [49] X. Z. Cai, G. Q. Wang, M. El Ghami, and Y. J. Yue, Complexity analysis of primaldual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term, Abstract and Applied Analysis. 11 pages, Art. ID 710158, 2014.
- [50] Y. E. Nesterov, A. S. Nemirovsky, Interior-polynomial methods in convex programming, vol. 13, Studies in applied mathematics, Society for Industrial and applied mathematics, Philadelphia, PA, 1994.
- [51] Y. Zhang, On the convergence of a class of infeasible-interior-point methods for the horizontal linear complementarity problem, SIAM J. Optim. 4; pp. 208-227, 1994.
- [52] Y. Zhang, On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming, SIAM J. Optim. 8; pp. 365-386, 1998.
- [53] 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.
- [54] Y. Q. Bai, M. El Ghami and C. Roos, A new efficient large-update primal-dual interior-point method based on a finite barrier, SIAM J. Optim. 13; pp. 766- 782, 2003.

- [55] Y. Q. Bai, M. El Ghami, C. Roos, A comparative study of kernel functions for primaldual interior point algorithms in linear optimization, *SIAM Journal on Optimization*. 15 ; pp. 101-128, 2004.
- [56] Y.Q. Bai, G. Lesaja, C. Roos, G.Q. Wang, and M. El Ghami. A class of large and small-update primal-dual interior-point algorithm for linear optimization. *Journal of optimization theory and application*. 138 (3), pp. 341-359, 2008.
- [57] Y.Q. Bai, J.L. Guo, C. Roos. A new kernel function yielding the best known iteration bounds for primal-dual interior-point algorithms. *Acta Mathematica Sinica. English Series*. 25 (12) ; pp. 2169-2178, 2009.
- [58] <http://infohost.nmt.edu/~sdplib/>.