

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS-SETIF

THÈSE

Présentée à la Faculté des Sciences
Département de Mathématiques
Pour L'obtention du Diplôme de

DOCTORAT SCIENCES

Option : Mathématiques appliquées

Par
Roumili Hayet

THÈME

**Méthodes de points intérieurs non réalisables en optimisation. Théorie,
Algorithme et Applications.**

Soutenue le : 10 / 12 /2007

Devant le jury :

Président :	Dr. N. Bensalem	Prof. Université Ferhat Abbas-Sétif (Algérie)
Directeurs de thèse :	Dr. A.Yassine	Prof. Université du Havre (France)
	Dr. A. Keraghel	M.C. Université Ferhat Abbas-Sétif (Algérie)
Examineurs :	Dr. J. Rodolphe Roche	Prof. Université de Nancy I (France)
	Dr. Sérigne Gueye	M.C. Université du Havre (France)
	Dr. M. Achache	M.C. Université Ferhat Abbas-Sétif (Algérie)

REMERCIEMENTS

J'ai effectué ma thèse sous la direction des professeurs A.Keraghel, Directeur du Laboratoire de Mathématiques Fondamentales et Numériques (LMFN) de l'Université Ferhat Abbas de Sétif en Algérie et A.Yassine, Directeur du Laboratoire de Mathématiques Appliquées du Havre (LMAH) en France. Je tiens à leur exprimer ma profonde reconnaissance. Leurs compétences, leurs conseils et leurs encouragements m'ont été très précieux.

Par ailleurs, je tiens à exprimer ma profonde gratitude, à Monsieur N. Bensalem, professeur à la Faculté des Sciences de l'Université Ferhat Abbas de Sétif, pour l'honneur qu'il me fait en présidant le jury.

Mes vifs remerciements s'adressent également à Messieurs :

Jean -Rodolphe Roche professeur à l'université de Nancy I, M. Achache maître de conférences à la Faculté des Sciences de l'Université Ferhat Abbas de Sétif et Sérigne Gueye maître de conférences à l'Université du Havre d'avoir accepté de juger ce travail et d'en être les examinateurs.

J'aimerais remercier également tous les membres de l'équipe d'optimisation à l'université de Sétif et tous les membres du Laboratoire de Mathématiques Appliquées du Havre (LMAH).

Je ne saurais oublier d'adresser un gentil mot à toute ma famille et mes amis qui m'ont beaucoup aidé.

ملخص

في هذه الدراسة نهتم بمسألة الانطلاق بالنسبة لطرق النقاط الداخلية غير المقبولة ذات المسار المركزي, آخذين كمرجع أبحاث Y.Zhang الخاصة بالبرمجة الخطية. بعد الانجاز العددي لخوارزمية مناسبة للبرمجة الخطية, نقترح تمديدا للبرمجة التربيعية المحدبة و آخرها للبرمجة نصف المعرفة.

الكلمات المفتاحية : طرق النقاط الداخلية، طرق المسار المركزي، البرمجة الخطية البرمجة التربيعية ، البرمجة النصف المعرفة.

Résumé

Dans cette étude, nous nous intéressons au problème d'initialisation dans les méthodes de points intérieurs non réalisables de type trajectoire centrale, en prenant comme référence les travaux de Y.Zhang pour la programmation linéaire (PL).

Après avoir mis en œuvre un algorithme approprié pour la programmation linéaire (PL), nous proposons une extension pour la programmation quadratique convexe (PQC) puis pour la programmation semi définie (SDP).

Mots clés : Méthodes de points intérieurs, méthodes de trajectoire centrale non réalisables, programmation linéaire (PL), programmation quadratique convexe (PQC), programmation semi définie positive (SDP).

Abstract

In this study, we are interested in the initialization problem of the infeasible interior point methods type central path following, taking Y.Zhang's work for the linear programming (PL) as references.

We just perform a suitable algorithm for linear programming and propose an extension of this algorithm for the convex quadratic programming and semidefinite programming.

Key words: Interior point methods, infeasible central path following methods, linear programming, quadratic convex programming, semidefinite programming

Table des matières

<i>Introduction générale.....</i>	<i>5</i>
-----------------------------------	----------

Chapitre I

Notions fondamentales

<i>I.1 : Préliminaires.....</i>	<i>9</i>
<i>I.2: Programmation mathématique (PM).....</i>	<i>11</i>
<i>I.2.1 : Classification d'un (PM).....</i>	<i>12</i>
<i>I.2.2: Qualification des contraintes</i>	<i>13</i>
<i>I.2.3: Principaux résultats d'existence</i>	<i>14</i>
<i>I.2.4 : Conditions d'optimalité</i>	<i>14</i>
<i>I.2.5 : Dualité lagrangienne.....</i>	<i>15</i>
<i>I.3 : Programmation linéaire (PL)</i>	<i>16</i>
<i>I.3.1 : Méthodes de résolutions d'un programme linéaire</i>	<i>19</i>
<i> Méthode du simplexe.....</i>	<i>19</i>
<i> Méthodes de points intérieurs.....</i>	<i>19</i>
<i> a) Méthode affine.....</i>	<i>20</i>
<i> b) Méthodes projectives avec potentiel.....</i>	<i>20</i>
<i> c) Méthodes de trajectoire centrale primales duales.....</i>	<i>21</i>

Chapitre II

Méthode de trajectoire centrale non réalisable pour la programmation linéaire (PL) et la programmation quadratique convexe (PQC)

<i>II.1 : Méthodes de points intérieur non réalisables pour la programmation linéaire (PL).....</i>	<i>27</i>
<i>II.1.1 : Principe général.....</i>	<i>27</i>
<i>II.1.2 : Description de l'algorithme....</i>	<i>28</i>
<i>II.1.3 : Principaux résultats de convergence.....</i>	<i>30</i>
<i>II.1.4: Détermination du pas de déplacement</i>	<i>31</i>
<i>II.1.5 : Simulations numériques</i>	<i>31</i>
<i>II.2 : Programmation quadratique convexe (PQC).....</i>	<i>38</i>
<i>II.2.1 : Introduction.....</i>	<i>38</i>
<i>II.2.2 : Dual d'un (PQC)</i>	<i>38</i>
<i>II.2.3 : Existence et unicité d'une solution optimale.....</i>	<i>39</i>
<i>II.2.4 : Conditions d'optimalité</i>	<i>40</i>
<i>II.2.5 : Méthodes de résolution d'un (PQC).....</i>	<i>40</i>
<i>II.2.6 : Méthode de trajectoire centrale non réalisable pour la (PQC).....</i>	<i>42</i>
<i>II.2.7 : Convergence de l'algorithme.....</i>	<i>44</i>
<i>II.2.8 : Simulations numériques</i>	<i>46</i>

Chapitre III

Méthodes de trajectoire centrale non réalisables pour la programmation semi définie (SDP)

<i>III.1 : Introduction</i>	50
<i>III.2 : Dualité et optimalité</i>	51
<i>III.3 : Présentation de la méthode</i>	56
<i>III.4 : Convergence de l'algorithme</i>	60
<i>III.5 : Commentaire</i>	63
Conclusion	65
Références	67

INTRODUCTION GÉNÉRALE

Introduction générale :

Les méthodes de points intérieurs sont apparues au milieu des années cinquante, générées essentiellement par la fonction barrière logarithmique proposée par Frisch [11]. Le terme point intérieur sera introduit dans les années soixante dans le fameux livre de Fiacco et McCormick [10].

Ces méthodes ont connu un succès théorique pendant quelques années, puis elles furent dominées par les méthodes de Lagrangien augmenté, du fait que ces dernières offrent plus d'opportunités algorithmiques.

Il faut reconnaître tout de même l'apport des méthodes de points intérieurs dans la théorie de la complexité algorithmique. Ce problème a débuté avec la programmation linéaire (*PL*) où, l'on a constaté une incohérence dans la méthode du simplexe: sa complexité algorithmique s'évalue théoriquement en $O(2^n)$ opérations, alors qu'en pratique le nombre d'opérations enregistré est souvent une fonction polynôme de la taille du problème.

En 1979 Khachiyan [15] introduit la méthode des ellipsoïdes qui donne lieu à un algorithme polynomial pour la programmation linéaire (*PL*), qui représente un succès théorique, cependant en pratique l'algorithme est loin de rivaliser avec l'approche simpliciale.

En 1984 Karmarkar [13] introduit une projective, de complexité polynomiale et compétitive numériquement avec la méthode du simplexe. Depuis, les méthodes de points intérieurs ont connu un développement exceptionnel, donnant lieu à un changement radical dans l'étude théorique et la résolution numérique des programmes mathématiques linéaires et non

linéaires. On parvient même à résoudre convenablement des problèmes non traitables auparavant.

Ces progrès rivalisant, n'échappent pas tout de même aux inconvénients théoriques, algorithmiques et numériques, entre autre, le problème d'initialisation qui devient de plus en plus difficile en passant aux problèmes non linéaires.

Ce problème d'initialisation est au centre de notre étude. Il figure depuis des années au menu de la recherche dans le domaine en programmation mathématique, pour atténuer à cet inconvénient, une alternative particulièrement intéressante pour nous est celle de trajectoire centrale non réalisable, introduite par Zhang (1994) [43] pour la programmation linéaire (*PL*). Il s'agit en fait de démarrer l'algorithme d'un point initial non nécessairement réalisable, entraîner ainsi une amélioration considérable du comportement numérique.

Nous avons repris différemment l'étude de Zhang, afin de tirer les conclusions nécessaires pour élaborer des extensions convenables pour la programmation quadratique convexe (*PQC*) et la programmation semi définie (*SDP*).

Concernant la programmation linéaire (*PL*), nos efforts sont concentrés sur l'aspect numérique: un algorithme de trajectoire centrale non réalisable est soigneusement implanté avec des aménagements pour le calcul de la direction et du pas de déplacement. Les résultats obtenus sont très instructifs quant aux performances réelles de la méthode.

Nous avons ensuite généralisé cet algorithme pour résoudre avec succès un problème quadratique convexe (*PQC*), les résultats obtenus sont encourageants, ce qui nous a motivé à considérer le problème semi défini

(*SDP*). L'algorithme ainsi proposé est cohérent et nous avons pu établir des résultats de convergence.

L'étude numérique et la mise en valeur de cet algorithme feront un bon sujet de recherche pour le futur.

La thèse est composée de trois chapitres. Le premier contient les notions préliminaires qui serviront d'appuis pour la suite. Le second chapitre, est consacré à la mise en œuvre de la méthode de trajectoire centrale non réalisable pour la programmation linéaire (*PL*) puis son extension pour la programmation quadratique convexe (*PQC*).

Enfin l'adaptation et l'étude théorique de l'algorithme de trajectoire centrale non réalisable pour la programmation semi-définie (*SDP*) sont présentées dans le dernier chapitre.

Mots clés : Méthodes de points intérieurs, méthodes de trajectoire centrale non réalisables, programmation linéaire (*PL*), programmation quadratique convexe (*PQC*), programmation semi-définie (*SDP*).

CHAPITRE I

Notions fondamentales

I.1: Préliminaires :

La notion de convexité est un outil mathématique important pour l'étude théorique et numérique des problèmes d'optimisation.

Définitions :

- Un ensemble D de \mathbb{R}^n est dit convexe si :

$$\forall x, y \in D: [x, y] = \{(1-\lambda)x + \lambda y, \lambda \in [0, 1]\} \subseteq D$$

- f (fonction réelle de $D \subset \mathbb{R}^n \rightarrow \mathbb{R}$) est dite convexe sur D si :

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

- f est dite strictement convexe sur D si :

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

- Un ensemble D de \mathbb{R}^n est dit affine si :

$$\forall x, y \in D \quad \text{et} \quad \lambda \in \mathbb{R}: (1-\lambda)x + \lambda y \in D$$

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

- On désigne par $B(0,1)$, la boule unité euclidienne de \mathbb{R}^n :

$$B(0,1) = \{ x \in \mathbb{R}^n : \|x\| \leq 1 \} = \{ x : d(x,0) \leq 1 \} \subset \mathbb{R}^n.$$

où $d(x,y)$ désigne la distance entre x et y .

$B(0,1)$ est un ensemble convexe fermé et borné et nous avons :

- $\forall a \in \mathbb{R}^n, \forall \varepsilon > 0$ la boule de centre a et de rayon ε s'écrit :

$$\{x : d(x, a) \leq \varepsilon\} = a + \{y : \|y\| \leq \varepsilon\} = a + \varepsilon B$$

- $C \subseteq \mathbb{R}^n$, l'ensemble des x dont la distance à C ne dépasse pas ε est :

$$\{x \in \mathbb{R}^n : \exists y \in C, \|x - y\| \leq \varepsilon\} = \cup \{y + \varepsilon B : y \in C\} = C + \varepsilon B.$$

La clôture (fermeture) de C , notée \bar{C} ou $cl(C)$ et son intérieur $\overset{\circ}{C}$ ou $int(C)$ peuvent donc s'écrire sous les formes suivantes :

$$cl(C) = \cap \{C + \varepsilon B / \varepsilon > 0\}$$

$$int(C) = \{x \in C / \exists \varepsilon > 0, x + \varepsilon B \subseteq C\}$$

- L'intérieur relatif d'un sous ensemble non vide C de \mathbb{R}^n noté :

$$ri(C) = \{x \in aff(C) / \exists \varepsilon > 0, (x + \varepsilon B) \cap aff(C) \subseteq C\}, \text{ il est défini comme}$$

l'intérieur de C vu comme sous ensemble de $aff(C)$.

- Une matrice carrée Q d'ordre n ($Q \in \mathbb{R}^{n \times n}$) est dite semi définie positive (resp. définie positive), et notée $Q \geq 0$ (resp. $Q > 0$), si pour tout vecteur $x \in \mathbb{R}^n$, on a $x^t Q x \geq 0$ (resp. $\forall x \neq 0, x^t Q x > 0$).

- Une matrice carrée Q d'ordre n ($Q \in \mathbb{R}^{n \times n}$) est dite semi définie négative (resp. définie négative), et notée $Q \leq 0$ (resp. $Q < 0$), si pour tout vecteur $x \in \mathbb{R}^n$, on a $x^t Q x \leq 0$ (resp. $\forall x \neq 0, x^t Q x < 0$).

- $f \in C^2$ est convexe (strictement convexe) si et seulement si le Hessien de f ($\nabla^2 f(x)$) est semi définie positive (définie positive) $\forall x$.

Particulièrement une forme quadratique $c'x + \frac{1}{2}x'Qx$ est convexe

(strictement convexe) si Q est semi définie positive (définie positive).

- f est dite fortement convexe sur IR^n s'il existe $\alpha > 0$ tel que :

$$\forall \lambda \in]0,1[, \forall x, y \in IR^n \quad f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y) - \frac{\alpha}{2} \lambda(1-\lambda) \|x-y\|^2$$

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

1.2: Programmation mathématique (PM).

Sans perte de généralités, nous considérons le programme mathématique (PM) suivant :

$$(PM) \quad \begin{cases} \min f(x) \\ x \in D \subseteq IR^n \end{cases}$$

où $D = U \cap V$, avec :

$$U = \{ x \in IR^n / f_i(x) \leq 0, i=1, \dots, m \} \text{ et } V = \{ x \in IR^n / g_i(x) = 0, i=1, \dots, p \}$$

f_i et g_i sont des fonctions de $IR^n \rightarrow IR$.

D est appelé ensemble des solutions réalisables et la fonction $f : IR^n \rightarrow IR$ est appelé fonction objectif (économique).

Définitions :

- On appelle solution réalisable de (PM) tout point vérifiant les contraintes (c. à. d : la solution appartient à D).
- Une solution réalisable qui optimise l'objectif sur D est dite solution optimale globale de (PM) et on note par $\operatorname{argmin}_D f(x)$ l'ensemble des solutions optimales globales.
- Un point $x^* \in D$ est une solution optimale locale de (PM) si : $\exists \mathcal{G}$ (voisinage) de x^* tel que : $f(x^*) \leq f(x); \forall x \in \mathcal{G}$ et on note par $\operatorname{locmin}_D f(x)$ l'ensemble des solutions optimales locales de (PM) et nous avons toujours $\operatorname{argmin}_D f(x) \subseteq \operatorname{locmin}_D f(x)$ de plus si (PM) est convexe (f et f_i sont convexes et g_i sont affines) les deux ensembles sont égaux.

Remarque 1:

Le programme mathématique (PM) précédent consiste :

- Soit à chercher un point optimal (local ou global).
- Soit à établir que $f(x)$ est non bornée inférieurement sur D , auquel cas on adopte la convention $\inf_D f(x) = -\infty$.
- Lorsque D est vide on pose par convention $\inf_D f(x) = +\infty$

1.2.1 : Classification d'un (PM).

On classe le problème (PM) à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité de la fonction objectif et les contraintes. A ce propos (PM) est dit convexe si f et f_i sont convexes et g_i sont affines et si ces dernières sont toutes différentiables, on dit que (PM) est différentiable.

La classe des (PM) convexes différentiables est le chapitre le mieux élaboré, les programmes non convexes ou non différentiables sont difficiles à traiter. Enfin, le cas le plus simple est celui de la programmation linéaire où f, f_i et g_i sont affines.

1.2.2 : Qualification des contraintes .

- Une contrainte d'inégalité $f_i(x) \leq 0$ ($\forall i$) est dite saturée (ou active) en $\bar{x} \in D$ si $f_i(\bar{x}) = 0$ ($\forall i$). Une contrainte d'égalité $g_i(x) = 0$ est par définition saturée en tout point x de D . Les contraintes sont qualifiées en $\bar{x} \in D$ si les gradients de toutes les fonctions des contraintes (f_i et g_i) saturées en \bar{x} forment un système linéairement indépendant.
- D est un polyèdre convexe (toutes les fonctions des contraintes sont affines) les contraintes sont qualifiées .
- D convexe (f_i convexes et g_i affines) et $\text{int}(D) \neq \emptyset$ les contraintes sont qualifiées.

Remarque 2:

La résolution complète de (PM) traite dans l'ordre les points suivants :

- L'existence (et éventuellement l'unicité) d'une solution optimale.
- Caractérisation de la solution (i.e. trouver des conditions d'optimalité).
- Elaboration d'algorithmes pour calculer numériquement cette solution.

I.2.3 : Principaux résultats d'existence.**Théorème 1 (Weierstrass) :**

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

Corollaire 1:

Si D est non vide et fermé et f est continue et coercive (au sens que $\lim_{\|x\| \rightarrow \infty} f(x) = \infty$) sur D alors (PM) admet au moins une solution optimale.

Remarque 3:

L'unicité d'une éventuelle solution est en général indépendante de l'existence, elle est souvent une conséquence de la convexité de D et de la stricte convexité de f .

I.2.4 : Conditions d'optimalité.

Reprenons le programme mathématique :

$$(PM) \quad \begin{cases} \min f(x) \\ f_i(x) \leq 0 & i = 1, \dots, m \\ g_i(x) = 0 & i = 1, \dots, p \\ x \in \mathbb{R}^n \end{cases}$$

Où f, f_i, g_i sont continûment différentiables.

Le lagrangien associé à (PM) est défini par

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^p \lambda_i g_i(x) + \sum_{i=1}^m \mu_i f_i(x)$$

Théorème 2 (Karush-Kuhn-Tucher (KKT)):

Soit $x^ \in D$ un point satisfaisant l'une des conditions de qualification précédentes alors il existe des multiplicateurs dits de Lagrange $\lambda \in \mathbb{R}^p$ et $\mu \in \mathbb{R}^m$ tel que:*

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \mu_i \nabla f_i(x^*) + \sum_{i=1}^p \lambda_i \nabla g_i(x^*) = 0 \text{ (conditions d'optimalité)} \\ \mu_i f_i(x^*) = 0 \quad i = 1, \dots, m \text{ (conditions de complémentarité)} \\ g_i(x^*) = 0 \quad i = 1, \dots, p \\ \mu_i \geq 0 \quad i = 1, \dots, m \end{cases}$$

Remarque 4:

- Si les contraintes ne sont pas qualifiées en x^* , les conditions de (KKT) ne s'appliquent pas (x^* peut être optimale sans vérifier ces conditions).

- Si (PM) est convexe, les conditions de (KKT) sont à la fois nécessaires et suffisantes pour que x^* soit optimum global.

I.2.5 : Dualité Lagrangienne.

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

$$(DM) \quad \begin{cases} \sup_{\lambda, \mu} \inf_{x \in D} L(x, \lambda, \mu) \\ \nabla_x L(x, \lambda, \mu) = 0 \\ \lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^m, \mu > 0 \end{cases}$$

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

I.3 : Programmation linéaire (PL).

On considère le programme linéaire primal (P) sous la forme standard suivante :

$$(P) \quad \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases}$$

où A est une matrice réelle de type $(m \times n)$ supposée de plein rang, $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

Le dual de (P) est donné par:

$$(D) \begin{cases} \max b^t y \\ A^t y \leq c \\ y \in \mathbb{R}^m \end{cases} \Leftrightarrow \begin{cases} \max b^t y \\ A^t y + z = c \\ y \in \mathbb{R}^m, z \geq 0 \end{cases}$$

Notations :

$S = \{ x \in \mathbb{R}^n : Ax = b, x \geq 0 \}$, l'ensemble des solutions réalisables de (P)
(c'est un polyèdre convexe et fermé de \mathbb{R}^n).

$S_{int} = \{ x \in \mathbb{R}^n : Ax = b, x > 0 \}$ L'ensemble des solutions strictement réalisables de (P).

$T = \{ y \in \mathbb{R}^m : A^t y \leq c, z \geq 0 \}$, l'ensemble des solutions réalisables de (D)
(c'est un polyèdre convexe et fermé \mathbb{R}^m).

- $T_{int} = \{ z \in \mathbb{R}^m, y \in \mathbb{R}^m : A^t y + z = c, z > 0 \}$ L'ensemble des solutions strictement réalisables de (D).

Théorème 3 (dualité faible) : [40]

Soient x et y deux solutions réalisables de (P) et (D) respectivement alors $b^t x \leq c^t y$. La quantité $c^t y - b^t x \geq 0$ s'appelle saut de dualité.

Théorème 4 (dualité forte) : [40]

Si x et y deux solutions réalisables de (P) et (D) respectivement tel que $b^t x = c^t y$ alors x et y sont des solutions optimales de (P) et (D).

Corollaire 2:

- Si $c^t x$ est non borné inférieurement sur S alors (D) est non réalisable ($T = \phi$).
- Si $b^t y$ est non borné supérieurement sur T alors (P) est non réalisable ($S = \phi$).
- Si (P) est non réalisable alors (D) est non borné et réciproquement.

Théorème 5 (des écarts complémentaires) : [40]

Soit \bar{x} et \bar{y} deux solutions réalisables primale et duale respectivement, posons $\bar{z} = A^t \bar{y} - c$ vecteur des variables d'écarts, alors \bar{x} et \bar{y} sont optimales si et seulement si $\bar{x}_i \bar{z}_i = 0, \forall i = 1, \dots, n$.

Proposition 1: [20]

Un programme linéaire réalisable et borné (objectif borné) possède au moins une solution optimale.

Définitions :

- Une base de A est une sous matrice B formée de m colonnes linéairement indépendantes de A .
- Soit B une base de A , la solution de base associée à B est le point $x = (x_B, x_N) \in \mathbb{R}^n$ tel que : $x_B = B^{-1}b, x_N = 0$.
- Une solution de base telle que $x_B \geq 0$ est dite solution de base réalisable.

- Un point $x \in S$ est un sommet de S si et seulement s'il est une solution de base réalisable.
- Un sommet est dit non dégénéré si $x_B > 0$. Il est dégénéré dans le cas contraire (au moins une composante de x_B est nulle).

Proposition 2: [14]

Si un programme linéaire possède une solution optimal, alors au moins un sommet des domaines réalisables est une solution optimale.

I.3.1 : Méthodes de résolution d'un programme linéaire.**Méthode du simplexe.**

Pour résoudre un (PL) une méthode tout à fait naturelle est celle du simplexe, développée par G. Dantzig [8]. Elle tient compte systématiquement des résultats établis précédemment. Elle évolue sur la frontière du domaine réalisable de sommet en sommet adjacent, en réduisant la valeur de l'objectif jusqu'à l'optimum. Un critère d'optimalité simple permet de reconnaître le sommet optimal. Le nombre de sommet étant fini, l'algorithme ainsi défini converge en un nombre fini d'itérations n'excédant pas le nombre $C_n^m = \frac{n!}{m!(n-m)!}$, sous l'hypothèse que tous les sommets visités sont non dégénérés. Dans le cas dégénéré, l'algorithme risque de cycler, cependant il existe des techniques convenables pour éviter ce phénomène.

En général la méthode du simplexe possède un comportement numérique très satisfaisant confirmé par ses applications multiples dans la résolution d'une large classe de problèmes pratiques. En théorie, la méthode n'a pas autant de succès elle est plutôt juger inefficace de par sa complexité arithmétique exponentielle de l'ordre de $O(2^n)$ opérations, un seuil atteint par les exemples de Klee et Minty [16].

Méthodes de points intérieurs.

Ces méthodes sont développées dans le but de résoudre convenablement des programmes mathématiques non linéaires, ayant des dimensions importantes. Certaines variantes sont conçues pour la (PL) histoire de retrouver une cohérence entre la théorie et la pratique, entre autre des algorithmes performants à complexité polynomiale.

Nous présentons sommairement dans ce paragraphe trois classes fondamentales de méthodes de points intérieurs.

a) Méthode affine.

C'est une méthode itérative due à Dikin [9] (1967) qui évolue dans l'intérieur relatif du domaine réalisable. En effet, à partir d'un point strictement réalisable ($x^0 > 0 / Ax^0 = b$), on construit une suite $\{x^k\}$ qui converge vers l'optimum sous certaines conditions. Le passage d'un itéré à l'autre se fait moyennant un déplacement convenable suivant une direction de descente calculée à partir d'un problème normalisé qui centre le point courant qu'on transforme ensuite dans l'espace originel. L'algorithme

correspondant à cette méthode est d'une structure simple, malheureusement la polynomialité est très difficile à établir.

b) Méthodes projectives avec potentiel.

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

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

La fonction objectif $f(x,y,z)$ est appelée fonction potentielle et $q > 0$ son paramètre, ces méthodes visent à ramener la valeur de $f(x,y,z)$ à $-\infty$. C'est ce type de fonction que Karmarkar introduit dans son article [13], la fonction potentielle est un intermédiaire pour réduire progressivement le saut de dualité à zéro. Ces méthodes ne sont pas aussi simples que les méthodes affines mais, on peut établir la polynomialité de l'algorithme correspondant.

c) Méthodes de trajectoire centrale primale duale (TC).

On s'intéresse à cette classe de méthodes qui ont été introduites à la même époque que les méthodes projectives avec potentielle (au début des années 90), elles possèdent des propriétés théoriques très intéressantes : complexité polynomiale, convergence asymptotique superlinéaire et éventuellement un statut numérique prometteur.

La trajectoire centrale du programme linéaire (PL) est définie à partir de la version barrière de (P) :

$$(P_\mu) \quad \min \left[c^t x - \mu \sum_{j=1}^n \log(x_j) \text{ ; } Ax = b, x > 0 \right]$$

où $\mu > 0$ est un paramètre barrière.

Proposition 3 : [20]

Si $S_{int} \neq \emptyset$, $T_{int} \neq \emptyset$ et A de plein rang, on a pour tout $\mu > 0$, le problème (P_μ) et son dual admettent des solutions optimales.

Hypothèse :

On suppose dans tout ce qui suit :

H_1 : $S_{int} \neq \emptyset$

H_2 : $T_{int} \neq \emptyset$.

Le programme (P_μ) étant convexe et différentiable, à contraintes linéaires (donc qualifiées partout), les conditions nécessaires et suffisantes d'optimalité de (KKT) s'écrivent :

$$\begin{cases} A^t y + \mu X^{-1} e - c = 0 \\ Ax = b \\ x > 0 \end{cases} \Leftrightarrow (1) \begin{cases} A^t y + z - c = 0 \\ Ax - b = 0 \\ XZe - \mu e = 0 \end{cases}$$

Où $X = \text{diag}(x_i)$, $X^{-1} = \text{diag}\left(\frac{1}{x_i}\right)$, $z = \mu X^{-1} e > 0$ et $Z = \text{diag}(z_i)$

En désignant par $(x(\mu), y(\mu), z(\mu))$ la solution du système non linéaire (1) pour $\mu > 0$ donné, l'ensemble :

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

est appelé trajectoire centrale de (P_μ).

D'après le système (1) le saut de dualité et le paramètre μ sont étroitement liés : $c^t x - b^t y = n\mu \Leftrightarrow \mu = x^t z / n$.

La stratégie des méthodes de trajectoire centrale, consiste à chercher des solutions approchées du système non linéaire (1), suivant la trajectoire centrale.

La version primale duale est plus intéressante que la version primale et la version duale pour les raisons suivantes:

- i)** Les résultats théoriques sont plus consistants.
- ii)** Les algorithmes sont plus performants en pratique.

Le système (1) est résolu par la méthode de Newton appliquée à $H(x, y, z) = 0$, où $H: \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$ définie par :

$$H(x, y, z) = \begin{bmatrix} A^t y + z - c \\ Ax - b \\ Xz - \hat{\mu} e \end{bmatrix}$$

pour chaque $\hat{\mu} = \sigma \mu$ ($0 < \sigma < 1$) et $(x, y, z) \in \overset{\circ}{S}$ (ensemble des solutions strictement réalisables de (P_μ)), on résout le système linéaire suivant:

$$\nabla H(x, y, z) \Delta w = H(x, y, z) \quad (2)$$

Avec : $\Delta w = (\Delta x, \Delta y, \Delta z)$ et la matrice jacobienne de H au point (x, y, z) définie par :

$$\nabla H(x, y, z) = \begin{pmatrix} 0 & A^t & I \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix}$$

Le nouvel itéré est alors défini par : $(\hat{x}, \hat{y}, \hat{z}) = (x, y, z) - (\Delta x, \Delta y, \Delta z)$

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

Facteur de centralité

Pour mesurer la qualité d'une solution trouvée, on introduit un facteur dit de " centralité " défini par le scalaire suivant: $\|XZe - \mu e\|$.

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

$$S_{cent}(\theta) = \{ (x,y,z) \in W / \|XZe - \mu e\| \leq \theta \mu : \mu > 0 \}.$$

où W est l'ensemble des solutions réalisables du (P) et son dual (D) .

Description de l'algorithme (TC).

Soient θ et δ deux constantes telles que:

$$0 \leq \theta < 1/2 \text{ et } 0 < \delta < \sqrt{n} \quad (\mathbf{a}_1)$$

$$(\theta^2 + \delta^2) / 2(1 - \theta) \leq \theta(1 - \delta/\sqrt{n}) \quad (\mathbf{a}_2)$$

Ces hypothèses permettent en particulier, de maintenir $x > 0$ et $z > 0$ au cours des itérés tout en gardant la solution trouvée toujours voisine de la trajectoire centrale [27].

L'algorithme correspondant est le suivant :

L'algorithme :**Début algorithme**

Données: soient $\varepsilon > 0$ un paramètre de précision fixé, θ et δ

deux constantes satisfaisant (\mathbf{a}_1) et (\mathbf{a}_2)

Initialisation: $(x^0, y^0, z^0) \in S_{cent}(\theta)$, $k = 0$,

Tant que : $(x^0)^t z^0 \geq \varepsilon$ faire

- a) $\mu_{k+1} = \mu_k(1 - \delta/\sqrt{n})$
- b) calculer $(\Delta x, \Delta y, \Delta z)$ solution du système d'équations (2) :
- c) calculer $x^{k+1} = x^k - \Delta x^k$, $y^{k+1} = y^k - \Delta y^k$ et $z^{k+1} = z^k - \Delta z^k$
- d) $k = k+1$

Fin tant que

Stop dès que : $(x^k)^t z^k < \varepsilon$

Fin algorithme.

Résultats de convergence pour la version primale duale.

Pour établir la convergence de cet algorithme nous avons besoin du théorème suivant :

Théorème 6 : [27]

Soient θ et δ deux constantes vérifiant les conditions **(a₁)** et **(a₂)**,

Erreur ! Signet non défini. > 0 défini par $\hat{\mu} = \mu(1 - \delta/\sqrt{n})$ où $\mu = (x^t z)/n$.

Supposons que : $(x, y, z) \in S_{cent}(\theta)$ et soit $\hat{w} = w - \Delta w$ tel que $w = (x, y, z)$ et $\hat{w} = (\hat{x}, \hat{y}, \hat{z})$ alors nous avons:

a) $\|\hat{X}\hat{Z}e - \hat{\mu}e\| \leq \theta\hat{\mu}$

b) $\hat{w} \in W$

c) $g(\hat{w}) = \hat{x}^t \hat{z} = n\hat{\mu}$

On peut généraliser les résultats de ce théorème pour toutes les itérés dans le corollaire suivant :

Corollaire 3: [27]

La suite de points w^k engendrée par l'algorithme satisfait :

$$a) \quad \|X^k Z^k e - \mu_k e\| \leq \theta \mu_k$$

$$b) \quad w^k \in W, \forall k = 1, \dots, n$$

$$c) \quad g(w^k) = (x^k)^t z^k = n\mu_k, k = 1, \dots, n \text{ où } \mu_k = \mu_0 (1 - \delta/\sqrt{n})^k.$$

La complexité arithmétique de l'algorithme est donnée par la proposition suivante :

Proposition 3: [27]

Le nombre total d'itérations effectuées par l'algorithme ne dépasse pas le nombre $K = \lceil (\sqrt{n}/\delta) \ln(n\varepsilon^{-1}\mu_0) \rceil$, avec une complexité arithmétique de l'ordre de $O(n^{3.5}L)$. L est une constante liée à la taille du problème primal(P).

Remarque 5 :

Signalons à la fin de ce chapitre que les méthodes de points intérieurs que nous venons de présenter, ont fait l'objet de plusieurs études de recherches relatives au coût élevé de l'itération et au problème d'initialisation, ce qui devient plus pesant au niveau des extensions pour la programmation non linéaire. Nous reviendrons sur ces points dans les chapitres qui suivent.

CHAPITRE II

*Méthodes (TC) non réalisable pour
(PL) et (PQC)*

II.1 : Méthode de trajectoire centrale non réalisable pour la programmation linéaire (PL).

Cette approche proposée par Zhang [43] pour la programmation linéaire en contournant la phase d'initialisation, on démarre d'un point intérieur non nécessairement réalisable et on tente de réaliser la faisabilité et l'optimalité via une certaine fonction de mérite (une fonction positive égale à zéro à l'optimum).

II.1.1 : Principe général.

Reprenons le programme paramétrisé (P_μ) associé au programme linéaire à résoudre (P) :

$$(P_\mu) \quad \min \left[c^t x - \mu \sum_{j=1}^n \log(x_j) \right], \quad Ax = b, \quad x > 0$$

Le système non linéaire de (KKT) correspondant au programme (P_μ) est donc le suivant :

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

Ce système sera résolu par la méthode de Newton à partir d'un point initial intérieur non nécessairement réalisable. Pour contrôler l'évolution des itérés on introduit la fonction de mérite définie comme suit :

$$\phi(x, y, z) = x^t z + r(x, y, z) \geq 0, \quad \forall (x, y, z) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n$$

où $r(x,y,z) = \|Ax - b\| + \|A^t y + z - c\|$. Il est clair que r mesure la faisabilité et $x^t z = c^t x - b^t y$ contrôle l'optimalité. L'idée est de faire tendre la valeur de ϕ vers zéro au cours des itérés.

II.1.2 : Description de l'algorithme.

Au début d'une itération $k > 0$ on dispose du point $(x, z) > 0, y \in \mathbb{R}^m$ tel que $\mu = x^t z / n > 0$, on détermine par la méthode de Newton le déplacement $(\Delta x, \Delta y, \Delta z)$ solution du système linéaire suivant :

$$(3) \quad \begin{bmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^t & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -XZe + \sigma\mu e \\ b - Ax \\ c - A^t y - z \end{bmatrix}$$

Le nouvel itéré est alors : $(\tilde{x}, \tilde{y}, \tilde{z}) = (x, y, z) + \alpha (\Delta x, \Delta y, \Delta z)$, avec $\alpha > 0$ est le pas de déplacement choisi de telle manière que $(\tilde{x}, \tilde{z}) > 0$ et $\phi(\tilde{x}, \tilde{y}, \tilde{z}) < \phi(x, y, z)$. Si la valeur de ϕ reste loin de zéro on réitère en remplaçant μ par $\tilde{\mu}$ ($\tilde{\mu} < \mu$).

La direction $(\Delta x, \Delta y, \Delta z)$ peut être calculée directement par l'application de la méthode d'élimination de Gauss au système précédent.

Les résultats obtenus sont valables mais limités aux petites dimensions, au sens que la matrice manipulée est de d'ordre $2n+m$.

Une alternative beaucoup plus économique consiste à réduire la taille de la matrice mise en œuvre, pour cela avec des simples calculs on obtient le système suivant :

$$(4) \begin{cases} (AZ^{-1}XA^t)\Delta y = -AZ^{-1}X(A^t y + z - c) + AZ^{-1}(Xz - \sigma\mu e) - Ax + b & (4a) \\ \Delta z = (A^t y + z - c) - A^t \Delta y \\ \Delta x = Z^{-1}(Xz - \sigma\mu e - X\Delta z) \end{cases}$$

Ce système d'équations est plus stable numériquement où la matrice mise en œuvre est d'ordre m , de plus elle est symétrique définie positive. Donc on applique la méthode de Cholesky pour résoudre le système.

On note par $\phi^k = \phi(x^k, y^k, z^k)$ la valeur de la fonction de mérite à la $k^{\text{ième}}$ itération et $r^k = r(x^k, y^k, z^k)$.

L'algorithme correspondant s'écrit :

Début algorithme (NRL)

Données : Soit $\varepsilon > 0$ un paramètre de précision fixé (suffisamment petit).

Initialisation : On choisi $(x^0, z^0) > 0$, $y^0 \in \mathbb{R}^m$ (arbitraires) et, $k = 0$

Tant que $\phi^0 > \varepsilon$ faire:

a) $\mu^k = (1/n)(x^k)^t z^k$ et choisir $\sigma^k \in (0,1)$

b) Résoudre le système (4), en calculant $\Delta x^k, \Delta y^k$ et Δz^k :

c) Trouver un pas de déplacement $\alpha^k > 0$ tel que : $x^{k+1} > 0$,

$z^{k+1} > 0$ et $\phi(x^{k+1}, y^{k+1}, z^{k+1}) < \phi(x^k, y^k, z^k)$

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

$k = k + 1$

Fin tant que

Stop dès que $\phi^k \leq \varepsilon$

Fin algorithme.

II.1.3 : Principaux résultats de convergence.

Sous les hypothèses (H1) et (H2), la convergence de l'algorithme est basée sur la proposition suivante:

Proposition 2: [43]

Si $0 < \alpha^k \leq 1$, la suite $\{\phi^k\}$ engendrée par l'algorithme satisfait:

$$\phi^{k+1} = (1 - \delta^k) \phi^k$$

$$\delta^k = \delta(\alpha^k) = [\alpha^k (1 - \sigma^k)(x^k)^t z^k + \alpha^k v^k r^0 - (\alpha^k)^2 (\Delta x^k)^t \Delta z^k] / [(x^k)^t z^k + v^k r^0].$$

La démonstration de cette proposition est basée sur le lemme suivant :

Lemme 1: [43]

Soit $\{(x^k, y^k, z^k)\}$ une suite générée par l'algorithme (NRL) alors:

$$1) A(x^k + \alpha^k \Delta x^k) - b = (1 - \alpha^k)(Ax^k - b) = v^{k+1}(Ax^0 - b)$$

$$2) A^t(y^k + \alpha^k \Delta y^k) + (z^k + \alpha^k \Delta z) - c = v^{k+1}(A^t y^0 + z^0 - c)$$

$$3) (x^k + \alpha^k \Delta x^k)^t (z^k + \alpha^k \Delta z^k) = (x^k)^t z^k (1 - \alpha^k + \alpha^k \sigma^k) + (\alpha^k)^2 (\Delta x^k)^t \Delta z^k$$

$$\text{où } v^{k+1} = (1 - \alpha^k) v^k = \prod_{j=0}^k (1 - \alpha_j) v^0, \quad v^0 = 1.$$

La fonction de mérite décroît d'une itération à l'autre d'un montant égal à $(1 - \delta^k)$, de plus on a le corollaire suivant :

Corollaire 1 : [43]

La suite $\{\phi^k\}$ converge au moins linéairement si $0 < \alpha^k \leq 1$ auquel cas nous avons $0 < \delta^k < 1$ et si δ^k tend vers 1 la convergence devient superlinéaire (comportement numérique).

La complexité arithmétique de l'algorithme est donnée par la proposition suivante :

Proposition 3: [40]

Si on considère le point initial $(x^0, y^0, z^0) = \zeta (e, 0, e)$ où $\zeta > 0$ alors: l'algorithme converge au bout de $O(n^2 |\log(\varepsilon)|)$ itérations (ε est un paramètre de précision)

II.1.4: Détermination du pas de déplacement [14].

Le choix du pas de déplacement a fait un axe de recherche important pendant de longues années, plusieurs procédures sont proposées, plus au moins équivalentes sur le plan numérique. Nous avons choisi ici, une technique simple et économique:

$$\alpha_x = \beta \alpha'_x \text{ et } \alpha_z = \beta \alpha'_z \text{ (} 0 < \beta < 1 \text{)}$$

où:

$$\alpha'_x = \begin{cases} \min \frac{-x_i}{\Delta x_i}, & i \in I \text{ et } I = \{ i \in \{1, \dots, n\} : \Delta x_i < 0 \} \\ 1 & \text{si } \Delta x_i \geq 0 \end{cases}$$

$$\alpha'_z = \begin{cases} \min \frac{-z_i}{\Delta z_i}, & i \in I \text{ et } I = \{ i \in \{1, \dots, n\} : \Delta z_i < 0 \} \\ 1 & \text{si } \Delta z_i \geq 0 \end{cases}$$

Cette alternative est réalisée avec succès dans notre implémentation numérique.

II.1.5 : Simulations numériques.

Les programmes sont réalisés en turbo-pascal sur un Pentium 4 avec une précision de l'ordre de 10^{-6} , on affichera les solutions optimales primales x^* duales y^* , la valeur optimale de la fonction objectif primale $f(\text{opt})$, la valeur optimale de la fonction objectif duale $g(\text{opt})$ et le nombre d'itérations (iter), pour différents types de programmes.

Remarque 1:

Si le problème initial est sous la forme canonique, on le ramène facilement à la forme standard en ajoutant des variables d'écart et artificielles.

Les exemples ci-dessous sont sous la forme canonique suivante:

$$\begin{cases} \min c'x \\ Ax \leq b \\ x \geq 0 \end{cases}$$

où A est une matrice réelle de type $(m \times n)$, $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

le dual est de la forme:

$$\begin{cases} \min b'y \\ A'y \geq c \\ y \in \mathbb{R}^m \end{cases}$$

Exemple 1:

$$b = (10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000)^t$$

$$c = (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1)^t$$

La matrice A est donnée par:

1	2	3	4	5	5	4	3	2	1
6	7	8	9	10	5	2	8	3	1
11	12	13	14	15	6	7	80	90	10
1	10	20	30	40	50	60	80	90	10
3	9	27	60	45	60	75	8	9	46
90	100	100	20	30	1000	900	25	1	1
3	30	300	2	20	200	1	10	100	150
5	10	15	20	25	30	35	40	45	50
1	11	111	2	22	222	3	33	333	4
7	70	8	80	9	90	10	100	15	155

La dimension du problème sous la forme standard est (10×20)

$$x^* = (50.147856, 0.000001, 0.000000, 18.005543, 155.920968, 0.000000, 0.000000, 15.981991, 17.503209, 31.899900)^t$$

$$y^* = (0.000000, 0.000000, 0.000000, -0.006307, -0.008124, -0.010587, -0.002302, 0.000000, -0.000296, -0.0013330)^t$$

$$f(\text{opt}) = -289.459468$$

$$g(\text{opt}) = -289.459440$$

$$\text{iter} = 11$$

Exemple 2:

$$b = (8, 4, 6, 2, 5, 1, 2, 6, 3, 9, 4)^t$$

$$c = (2, -1, -3, 5, -2, 0, 4, 1, 2, -1, 1, -1, 0, 2)^t$$

les élément de la matrice A sont représentés par le tableau suivant:

1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	2	1	0	0	0	0	0	0	0	0	0	0	0
0	0	3	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	2	4	0	0	0	0	0	0	0	0	0
0	0	0	0	6	2	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	2	0	0	0	0	0	0	0
0	0	0	0	0	0	4	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	1	0	0	0	0
0	0	0	0	0	0	0	0	3	-1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	2	0	0
0	0	0	0	0	0	0	0	0	0	0	1	2	1

La dimension du problème sous la forme standard est (11×25)

$$x^* = (0.000000, 1.000000, 2.000000, 0.000000, 0.500000, 0.474222, 0.000000, 0.000000, 0.000000, 1.500000, 0.000000, 3.750000, 0.057435, 0.000000)^t$$

$$y^* = (0.000000, -0.500000, -0.833333, -0.500000, 0.000000, 0.000000, 0.000000, -0.250000, 0.250000, -0.500000, 0.000000)^t$$

$$f(\text{opt}) = -13.250000$$

$$g(\text{opt}) = -13.250000$$

$$\text{iter} = 12$$

Exemple 3:

$$b = (420, 415, 355, 345, 160, 95, 380, 395, 270, 230, 310, 420, 5200, 5200, 3600, 0)^t$$

$$c = (-139, -88, -133, -137, -165, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^t$$

La matrice A est donnée par:

1.4	1.8	1.5	1.4	1.4	0	0	0	0	0	0
9.8	2.4	1.4	1.4	1.4	0	0	0	0	0	0
4	0.4	1.4	1.4	1.4	0	0	0	0	0	0
2.8	0.6	1.3	1.4	1.5	5.5	0	0	0	0	0
2.2	0.4	1.3	1.5	1.2	0	5.5	0	0	0	0
2.2	0.4	1.3	1.3	1.2	0	0	5.5	0	0	0
2.2	0.6	1.3	1.3	1.2	0	0	0	5.5	0	0
2.6	5.8	1.5	1.5	1.2	0	0	0	0	5.5	0
0.6	4	1.3	0	0	0	0	0	0	0	5.5
0.6	1.2	1.3	1.3	2.6	0	0	0	0	0	0
0.6	1.8	1.2	1.2	1.2	0	0	0	0	0	0
0.6	1.8	1.5	1.4	1.4	0	0	0	0	0	0
16	0	0	12	35	50	50	0	0	0	0
20	0	0	36	50	0	0	50	50	0	0
16	0	0	12	35	0	0	0	0	50	50
0	0.1	0.9	0.8	2.3	-1	-1	-1	-1	-1	-1

La dimension du problème sous la forme standard est (16×27) .

$$x^* = (0.000000, 54.801584, 0.000000, 20.357231, 38.845805, 40.972767, 11.0777919, 0.000000, 49.825397, 0.000000, 9.235212)^t$$

$$y^* = (0.000000, 0.000000, 0.000000, -4.594670, -4.594670, -68.559607, -4.594670, -5.572230, -4.594670, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, -25.270687)^t$$

$$f(\text{opt}) = -14.21.037868$$

$$g(\text{opt}) = -14021.037868, \text{ iter} = 17$$

Exemple 4:

Ce problème est sous la forme:

$$\begin{cases} \min & c^t x \\ Ax & \geq b \\ x & \geq 0 \end{cases}$$

$$b = (2300, 75, 38, 660, 1300, 10, 1900, 1.2, 1.2, 63, 400)^t$$

$$c = (25, 30, 350, 150, 40, 20, 100, 40, 60, 100, 17, 20, 20, 12, 75, 900, 20)^t$$

A est donnée par:

351	270	260	451	156	59	721	58	130	118	77	51	40	24	28	311	40
6.2	8	17.5	12	12.7	2.9	0.6	6	17.5	20	1.9	1.3	1.2	1.6	3	34.2	0.8
0.8	1.5	20.5	44.3	11.2	3.3	81.6	3.5	6	3.5	0.1	0.2	0.2	0.2	0.4	0.7	0.3
6	11	6	9	65	100	10	120	80	12	5	35	40	45	98	470	14
2	480	90	90	90	36	780	5	100	90	12	57	10	15	25	600	4
0.4	1	1.3	1.2	2.6	0.1	0.1	1.4	3	0.7	0.5	0.5	0.5	0.4	3.3	23	0.2
0	0	25	0	800	120	2400	0	60	40	0	1300	6	33	2600	1000	40
0.09	0.1	0.04	0.4	0.1	0.04	0.01	0.02	0.02	0.15	0.1	0.06	0.03	0.08	0.12	0.21	0.09
0.03	0.03	0.11	0.1	0.4	0.15	0.03	0.02	0.15	0.2	0.03	0.04	0.02	0.05	0.3	1	0.02
0	0	0	0	0	2	0	0	1	1	15	7	10	50	100	20	50
0	0	0	0	10	0	0	0	530	0	0	0	0	0	0	0	0

La dimension de ce problème sous la forme standard est (11×28) .

$$x^* = (3.450965, 1.770912, 0.000000, 0.000000, 0.755766, 2.071795, \\ 0.140651, 0.000000, 0.740457, 0.000000, 0.000000, 0.347382, \\ 6.460187, 0.000000, 0.000000, 0.000000)^t$$

$$y^* = (0.041955, 0.610660, 0.509866, 0.103865, 0.012174, 0.000000, \\ 0.006934, 60.358619, 0.000000, 0.000000, 0.055944)^t$$

$$f(\text{opt}) = 354.030419$$

$$g(\text{opt}) = 354.030419$$

$$\text{iter} = 12$$

Exemple 5:

On présente ici un exemple où l'ensemble des solutions strictement réalisables ($x_i > 0 \forall i$) est vide auquel cas les méthodes de points intérieurs réalisables ne sont pas applicables, cependant avec l'algorithme non réalisable on parvient à obtenir convenablement les solutions optimales.

$$\begin{cases} \min x_1 - 3x_2 \\ x_1 + x_2 + x_3 = 5 \\ x_1 + x_2 = 5 \\ x_i \geq 0 \quad \forall i = 1,2,3 \end{cases}$$

$$x^* = (0.000000, 5.000000, 0.000000)^t$$

$$y^* = (-203.627267, 200.627267)^t$$

$$f(\text{opt}) = -15$$

$$g(\text{opt}) = -15$$

$$\text{iter} = 3$$

II.2 : Programmation quadratique convexe (PQC).

II.2.1 : Introduction.

La programmation quadratique est un chapitre bien élaboré du domaine de l'optimisation. Elle est connue par ses applications multiples dans les domaines scientifique et pratiques. Sur le plan algorithmique, souvent on fait intervenir des programmes quadratiques comme procédures intermédiaires pour des programmes non linéaires, c'est le cas entre autre des méthodes de programmation quadratique successive.

Sans perte de généralité, on peut présenter un programme quadratique sous la forme suivante :

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

Où Q est une matrice symétrique d'ordre n , $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $A \in \mathbb{R}^{m \times n}$ de plein rang .

Rappelons que l'ensemble des contraintes $S = \{ x \in \mathbb{R}^n : Ax = b, x \geq 0 \}$ est un polyèdre convexe fermé, la fonction objectif est indéfiniment différentiable est nous avons en particulier :

$$\begin{aligned} \nabla f(x) &= Qx + c \\ \nabla^2 f(x) &= Q \end{aligned}$$

(PQ) est convexe $\Leftrightarrow f$ est convexe $\Leftrightarrow Q$ est semi définie positive.

On s'intéresse dans cette partie à la résolution d'un programme quadratique convexe (PQC), supposant qu'il admet au moins une solution optimale :

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

où Q est une matrice symétrique semi définie positive d'ordre n , $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $A \in \mathbb{R}^{m \times n}$ de plein rang ($rg(A) = m \leq n$).

II.2.2 : Dual d'un (PQC).

Le dual de (PQC) s'obtient très facilement, il s'écrit sous la forme :

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

où $y \in \mathbb{R}^m$ désigne la variable duale et $z \in \mathbb{R}^n$ est le vecteur de variable d'écart unique.

II.2.3 : Existence et unicité d'une solution.

Les résultats classiques (Weierstrass et corollaire 1 du chapitre I) s'appliquent et nous avons de plus le résultat suivant :

Lemme 2:

Si Q est définie positive et S est non vide alors (PQC) admet une solution optimale unique

II.2.4 : Conditions d'optimalité.

Les contraintes de (PQC) sont qualifiées puisque linéaires, on peut donc écrire les conditions d'optimalités de (KKT) qui sont nécessaires et suffisantes.

$$\begin{cases} c + Qx - A^t y - z = 0 \\ Ax = b \\ x^t z = 0 \\ x \geq 0, z \geq 0 \end{cases}$$

II.2.5 : Méthodes de résolution d'un (PQC).

On peut classifier les méthodes de résolutions en trois catégories conformément à leurs principes.

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 : soit x^0 un point satisfaisant les contraintes ($Ax^0 = b$) et posons $x = x^0 + P_\Lambda y$ avec $P_\Lambda = I - A^T(AA^T)^{-1}A$ l'opérateur de projection sur le noyau de A ($Ker(A)$). 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 Q de la forme quadratique : $d_i Q d_j = 0 \quad \forall i \neq j, i, j \in \{0, 1, \dots, k\}$

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 que (PQC) de la forme :

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

Où f est différentiable non nécessairement convexe.

2/ Méthodes simpliciales.

Parmi les méthodes simpliciales on cite, celle de gradient réduit dûe à Wolfe. C'est une extension directe de la méthode du simplexe, appliquée à (PQC). De ce fait elle présente les mêmes inconvénients à savoir cyclage et 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 programme quadratique convexe. Ce sont des extensions des méthodes développées pour la programmation linéaire (projectives et de trajectoire centrale). Les problèmes d'initialisation, le coût de l'itération et le choix de la direction de déplacement deviennent plus pesants. A ce propos, nous proposons une variante de trajectoire centrale non réalisable.

II.2.6 : Méthodes de trajectoire centrale non réalisable pour la (PQC).

Reprenons le programme quadratique convexe (PQC):

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

où Q est une matrice symétrique semi définie positive d'ordre n , $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $A \in \mathbb{R}^{m \times n}$ de plein rang ($rg(A) = m \leq n$).

le dual (DQC):

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

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

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

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

Le principe est le même que pour la programmation linéaire (PL), on résout le système non linéaire de (KKT) ci-dessous (5) associé au problème (PQC_μ) par la méthode de Newton, en partant d'un point intérieur non nécessairement réalisable.

$$(5) \begin{cases} c + Qx - \mu X^{-1} e - A^t y = 0 \\ Ax = b \\ x > 0 \end{cases} \Leftrightarrow (6) \begin{cases} c + Qx - z - A^t y = 0 \\ Ax = b \\ Xz = \mu e \\ x > 0, z > 0 \end{cases}$$

Pour contrôler la faisabilité et l'optimalité, on introduit la fonction de mérite suivante :

$$\phi(x, y, z) = x^t z + \|Ax - b\| + \|-Qx + A^t y + z - c\|$$

On résout le système d'équations non linéaire (6) à partir d'un point initial $x > 0, y \in \mathbb{R}^m, z > 0$ et $\mu = x^t z / n > 0$ par la méthode de Newton, on obtient le système suivant:

$$(7) \begin{cases} Q\Delta x - \Delta z - A^t \Delta y = -c - Qx + z + A^t y \\ A\Delta x = b - Ax \\ X\Delta z + Z\Delta x = -XZe + \sigma \mu e \end{cases}$$

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

Le nouvel itéré est : $(\tilde{x}, \tilde{y}, \tilde{z}) = (x, y, z) + \alpha (\Delta x, \Delta y, \Delta z)$ où $\alpha > 0$ est le pas de déplacement choisi de telle manière que $(\tilde{x}, \tilde{z}) > 0$ et

$\phi(\tilde{x}, \tilde{y}, \tilde{z}) < \phi(x, y, z)$. Si la valeur de ϕ reste loin de zéro on remplace μ par $\hat{\mu}$ ($\hat{\mu} < \mu$) et on réitère.

On obtient le schéma algorithmique suivant:

Algorithme (NRQ).

Début algorithme

Données : Soit $\varepsilon > 0$ un paramètre de précision fixé (suffisamment petit).

Initialisation : On choisi $(x^0, z^0) > 0, y^0 \in \mathbb{R}^m$ (arbitraires) et, $k = 0$

Tant que $\phi^0 > \varepsilon$ faire:

a) $\mu^k = (1/n)(x^k)^t z^k$ et choisir $\sigma^k \in (0,1)$

b) Résoudre le système (7), en calculant $\Delta x, \Delta y$ et Δz :

c) Trouver un pas de déplacement $\alpha^k > 0$ tel que : $x^{k+1} > 0,$

$$z^{k+1} > 0 \text{ et } \phi(x^{k+1}, y^{k+1}, z^{k+1}) < \phi(x^k, y^k, z^k)$$

$$\text{où } (x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha^k(\Delta x, \Delta y, \Delta z)$$

d) $k = k + 1$

Fin tant que

Stop dès que $\phi^k \leq \varepsilon$

Fin algorithme.

II.2.7 : Convergence de l'algorithme.

On suppose que les ensembles des solutions réalisables des problèmes (PQC) et (DQC) sont respectivement non vide, alors on a les résultats suivants :

Proposition 4:

Si $0 < \alpha^k \leq 1$, la suite $\{\phi^k\}$ engendrée par l'algorithme satisfait:

$$\phi^{k+1} = (1 - \delta^k) \phi^k$$

où $\delta^k = \delta(\alpha^k) = [\alpha^k (1 - \sigma^k)(x^k)^t z^k + \alpha^k v^k r^0 - (\alpha^k)^2 (\Delta x^k)^t \Delta z^k] / [(x^k)^t z^k + v^k r^0]$.

La démonstration de cette proposition est basée sur le lemme suivant :

Lemme 3:

Soit $\{(x^k, y^k, z^k)\}$ une suite générée par l'algorithme (NRQ) alors:

- 1) $A(x^k + \alpha^k \Delta x^k) - b = (1 - \alpha^k)(Ax^k - b) = v^{k+1}(Ax^0 - b)$
- 2) $A^t(y^k + \alpha^k \Delta y^k) + (z^k + \alpha^k \Delta z^k) - c - Q(x^k + \alpha^k \Delta x^k) = v^{k+1}(A^t y^0 + z^0 - Qx^0 - c)$
- 3) $(x^k + \alpha^k \Delta x^k)^t (z^k + \alpha^k \Delta z^k) = (x^k)^t z^k (1 - \alpha^k + \alpha^k \sigma^k) + (\alpha^k)^2 (\Delta x^k)^t \Delta z^k$

$$\text{où } v^{k+1} = (1 - \alpha^k) v^k = \prod_{j=0}^k (1 - \alpha^j) v^0, \quad v^0 = 1.$$

La fonction de mérite décroît d'une itération à l'autre d'un montant égal à $(1 - \delta^k)$, de plus on a le corollaire suivant :

Corollaire 2:

La suite $\{\phi^k\}$ converge au moins linéairement si $0 < \alpha^k \leq 1$ auquel cas nous avons $0 < \delta^k < 1$ et si δ^k tend vers 1 la convergence devient superlinéaire.

II.2.8 : Simulations numériques.

Exemple 1 :

$$\left\{ \begin{array}{l} \min_{(x,t)} f(x,t) = 6.5x + 0.5x^2 - t_1 - 2t_2 - 3t_3 - 2t_4 - t_5 \\ Az \leq b, z = (x,t)^t \\ 0 \leq t_i \leq 1 \quad i = 3,4 \\ 0 \leq t_5 \leq 2 \end{array} \right.$$

$$b = (26, -11, 24, 12, 3)^t$$

$$\begin{pmatrix} 1 & 2 & 8 & 1 & 3 & 5 \\ -8 & -4 & -2 & 2 & 4 & -1 \\ 2 & 0.5 & 0.2 & -3 & -1 & -4 \\ 0.2 & 2 & 0.1 & -4 & 2 & 2 \\ -0.1 & -0.5 & 2 & 5 & -5 & 3 \end{pmatrix}$$

$$z^* = (x^*, t^*)^t = (0, 7.987342, 0.253165, 2, 2, 0)^t$$

$$y^* = (-0.246835, 0, 0, -0.253165, 0)^t$$

$$f(\text{opt}) = -18.493672$$

$$g(\text{opt}) = -18.483671$$

$$\text{iter} = 12$$

Exemple 2 :

$$\left\{ \begin{array}{l} \min f(x) = \sum_{i=1}^9 x_i x_{i+1} + \sum_{i=1}^8 x_i x_{i+2} + x_1 x_9 + x_1 x_{10} + x_2 x_{10} + x_1 x_5 + x_4 x_7 \\ \sum_{i=1}^{10} x_i = 1, \quad x_i \geq 0 \quad \forall i \end{array} \right.$$

$$x^* = (0, 0.249335, 0.25, 0, 0.000665, 0.017431, 0, 0.25, 0.232568, 0)^t$$

$$y^* = 0.25$$

$$f(\text{opt}) = 0.125010$$

$$g(\text{opt}) = 0.125000, \text{ iter} = 5$$

Exemple 3 : $m = 3, n = 5$

$$A = \begin{pmatrix} 1 & 1.2 & 1 & 1.8 & 0 \\ 3 & -1 & 1.5 & -2 & 1 \\ -1 & 2 & -3 & 4 & 2 \end{pmatrix}, Q = \begin{pmatrix} 20 & 1.2 & 0.5 & 0.5 & -1 \\ 1.2 & 32 & 1 & 1 & 1 \\ 0.5 & 1 & 14 & 1 & 1 \\ 0.5 & 1 & 1 & 15 & 1 \\ -1 & 1 & 1 & 1 & 16 \end{pmatrix}, c = \begin{pmatrix} 1 \\ -15 \\ 2 \\ 15 \\ 3 \end{pmatrix}, b = \begin{pmatrix} 9.31 \\ 5.45 \\ 6.60 \end{pmatrix}$$

$$x^* = (2.632276, 0.701827, 1.399507, 2.464458, 1.084655)^t$$

$$y^* = (25.271033, 11.733714, 5.257142)^t$$

$$f(\text{opt}) = 172.733206$$

$$g(\text{opt}) = 172.733196$$

$$\text{iter} = 28$$

Exemple 4 : $m = 3, n = 10$

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix}, b = \begin{pmatrix} 11651 \\ 16672 \\ 21295 \end{pmatrix}$$

$$c = (-0.5, -1, 0, 0, -0.5, 0, 0, -1, -0.5, -1)^t$$

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

$$x^* = (0.963886, 0.509607, 1.739953, 1.905056, 1.243511, 2.626820, \\ 1.322918, 1.617087, 0.824013, 0.897582)^t$$

$$y^* = (4.243380, 22.362785, 5.192083)^t.$$

$$f(\text{opt}) = 264.148690$$

$$g(\text{opt}) = 264.148699$$

$$\text{iter} = 27$$

Commentaire :

Les résultats obtenus pour la programmation linéaire et la programmation quadratique convexe sont très satisfaisants et encourageants.

CHAPITRE III

*Méthode (TC) non réalisable pour
(SDP)*

III.1 : Introduction.

Le problème de la programmation semi définie (SDP) linéaire, est connu depuis le début des années 60 comme un problème d'optimisation (convexe) lié à des applications : valeurs propres extrémales d'une matrice, optimisation combinatoire etc...

Il s'écrit sous la forme :

$$(PSDP) \begin{cases} \min_{X \in S^n} C \bullet X = tr(CX) = \sum_i \sum_j C_{ij} X_{ij} \\ A_i \bullet X = b_i, i = 1, \dots, m \\ X \geq 0 \end{cases}$$

Où $b_i \in \mathbb{R}$ et $C, A_i \in S^n$ (espace des matrices symétriques), le symbole tr désigne la trace d'une matrice

C'est une généralisation de la programmation linéaire (PL) au sens que, à la place du vecteur (inconnu) $x \in \mathbb{R}_+^n$ on a une $(n \times n)$ matrice X symétrique semi définie positive, le coût est remplacé par la trace de CX .

Donc un programme linéaire et un programme semi défini positif où X désigne la matrice diagonale des x_i ($X = \text{diag}(x_i)$) et $C = \text{diag}(c_i)$

De ce fait la théorie de la programmation linéaire est transportée sans peine à (SDP), à quelques légères différences au niveau des résultats de dualité.

L'aspect algorithmique pose en revanche des difficultés majeures faisant de l'implantation numérique, un véritable challenge à réaliser. A ce propos, toute extension de la méthode simpliciale est exclue du fait que l'ensemble des contraintes est un cône non polyédral dépourvu de sommets.

Au début des années 90, grâce notamment aux travaux fondateurs de

ALIZADEH, NEMIROVSKI, NESTEROV et autres, le problème (SDP) est relancé pour subir un traitement décent via l'approche de points intérieurs, allant de l'aspect théorique pur à celui de la mise en œuvre proprement dite.

Tout particulièrement, la méthode de la trajectoire centrale présente plusieurs atouts en faveur d'un traitement algorithmique convenable.

Le problème d'initialisation devient alors plus difficile à traiter. Nous proposons dans ce chapitre une variante de points intérieurs non réalisables pour surmonter cette difficulté. On établira les résultats de convergence de l'algorithme proposé.

III.2 : Dualité et optimalité.

Reprenons le problème :

$$(PSDP) \begin{cases} \min_{X \in S^n} C \bullet X \\ A_i \bullet X = b_i, i = 1, \dots, m \\ X \geq 0 \end{cases}$$

De la même manière que pour les programmes linéaires on peut trouver le dual du (PSDP), on associe aux contraintes $A_i \bullet X = b_i$ les variables $y_i \in \mathbb{R}, i = 1, \dots, m$, on forme alors la fonction lagrangienne :

$$L(X, y) = C \bullet X + \sum_{i=1}^m y_i (b_i - A_i \bullet X)$$

on minimise la fonction $L(x, y)$ par rapport à X :

$$\begin{aligned} \min_{X \in S^n, X \geq 0} L(X, y) &= \min_{X \in S^n, X \geq 0} [C \bullet X + \sum_{i=1}^m y_i (b_i - A_i \bullet X)] \\ &= \min_{X \in S^n, X \geq 0} [C \bullet X + \sum_{i=1}^m y_i b_i - \sum_{i=1}^m y_i (A_i \bullet X)] \\ &= \min_{X \in S^n, X \geq 0} [b^t y + (C - \sum_{i=1}^m y_i A_i) \bullet X] / y = (y_1, \dots, y_m)^t, b = (b_1, \dots, b_m)^t \end{aligned}$$

ce problème n'a pas de solution que si $(C - \sum_{i=1}^m y_i A_i) \geq 0$.

En effet, si $(C - \sum_{i=1}^m y_i A_i) \leq 0$ il est possible de trouver un $X \geq 0$ tel que la quantité $(C - \sum_{i=1}^m y_i A_i) \bullet X$ soit aussi négative que l'on veut, le minimum ne peut être alors que $-\infty$.

En introduisant la nouvelle variable dual $Z = (C - \sum_{i=1}^m y_i A_i)$, la fonction duale au sens de Lagrange devient :

$$\mathfrak{J}(y) = \max_{y \in \mathbb{R}^m, Z \geq 0} [b^t y + \min_{X \geq 0} Z \bullet X]$$

avec :

$$\min_{X \in S^n, X \geq 0} Z \bullet X = \begin{cases} 0 & \text{si } Z \geq 0 \\ -\infty & \text{si non} \end{cases}$$

On peut alors montrer que le problème dual s'écrit :

$$(DSDP) \begin{cases} \max_{y \in \mathbb{R}^m, Z \in S^n} b^t y \\ \sum_{k=1}^m y_k A_k + Z = C \\ Z \geq 0 \end{cases}$$

Notons que le dual à une formulation plus simple que le primal, ceci peut être avantageux pour développer des méthodes duales.

Notations :

Notons par :

$K = \{ X \in S^n / A_i \bullet X = b_i, i=1, \dots, m, X \geq 0 \}$ l'ensemble des solutions réalisables de (PSDP) .

$K_{int} = \{ X \in S^n / A_i \bullet X = b_i, i=1, \dots, m, X \succ 0 \}$ l'ensemble des solutions

strictement réalisables de (PSDP) .

$U = \{ y \in \mathbb{R}^m, Z \in \mathcal{S}^n / \sum_{k=1}^m y_k A_k + Z = C, Z \geq 0 \}$ l'ensemble des solutions réalisables de (DSDP)

$U_{int} = \{ y \in \mathbb{R}^m, Z \in \mathcal{S}^n, / \sum_{k=1}^m y_k A_k + Z = C, Z \succ 0 \}$ l'ensemble des solutions strictement réalisables de (DSDP)

Les résultats de dualité faible de la programmation linéaire (PL) s'étendent sans peine aux problèmes (SDP).

Lemme 1:

Soient $X \in K$ et $y \in U$ deux solutions réalisables de (PSDP) et (DSDP) respectivement alors, $b^t y \leq C \bullet X$.

La quantité $C \bullet X - b^t y \geq 0$ s'appelle saut de dualité.

Preuve :

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

Puisque les matrices $(C - \sum_{i=1}^m y_i A_i)$ et X sont les deux semi définies positives, la

trace de leur produit est un nombre positif ou nul, i.e., $\left\langle C - \sum_{i=1}^m y_i A_i, X \right\rangle \geq 0$, ce

qui donne $b^t y \leq C \bullet X$.

Concernant la dualité forte, l'existence de solutions strictement réalisables primales et duales est nécessaire comme le montre le lemme suivant :

Lemme 2 :[39]

Soient X et y deux solutions réalisables de (PSDP) et (DSDP) respectivement et supposons $K_{int} \neq \emptyset$ ou $U_{int} \neq \emptyset$, alors X et y sont des solutions optimales si et seulement si $(b^t y - C \cdot X) = X \cdot Z = 0$.

Posons une fois pour toutes, les hypothèses suivantes:

- (H1): $K_{int} \neq \emptyset$
- (H2): $T_{int} \neq \emptyset$
- (H3): A_i sont linéairement indépendantes, $\forall i = 1, \dots, m$.

Nous avons besoin du résultat suivant :

Lemme3 :

Soit $\tilde{\mathcal{M}}_n(\mathbb{R})$ l'espace des $n \times n$ matrice réelles muni du produit scalaire : $\langle A, B \rangle = \text{tr}(A^t B)$ et Ω un ouvert des matrices inversibles de $\tilde{\mathcal{M}}_n(\mathbb{R})$.

Soit $L : \tilde{\mathcal{M}}_n(\mathbb{R}) \rightarrow \mathbb{R}$ une forme linéaire, alors L est continue et

$\exists M \in \tilde{\mathcal{M}}_n(\mathbb{R})$ tel que $L(H) = \langle M, H \rangle = \text{tr}(M^t H) \quad \forall H \in \tilde{\mathcal{M}}_n(\mathbb{R})$.

Si $L = \mathcal{D}f(x^0) : f : \Omega \rightarrow \mathbb{R} \Rightarrow M = \nabla f(x^0)$

Différentiabilité de quelques fonctions de matrices

1/ La trace :

$$tr : A \in \widetilde{\mathcal{M}}_n(\mathbb{R}) \rightarrow tr A \in \mathbb{R}$$

$tr A = \langle I, A \rangle = tr(I^t A)$ c'est une application linéaire (continue) donc sa

différentielle en tout $A \in \widetilde{\mathcal{M}}_n(\mathbb{R})$ est $\mathcal{D} tr(A) : H \rightarrow tr(H) = \langle I, H \rangle$

$$d'où \nabla tr(A) = \nabla tr(I^t A) = I.$$

2/ le déterminant (det) :

$det : A \in \widetilde{\mathcal{M}}_n(\mathbb{R}) \rightarrow det A \in \mathbb{R}$ une application multilinéaire (continue)

donc différentiable en tout $A = [a^1, a^2, \dots, a^n]$ où a^i sont les colonnes de A

$$\mathcal{D} det(A) : H = [h^1, h^2, \dots, h^n] \rightarrow \sum_{j=1}^n (dét [a^1, a^2, \dots, a^{j-1}, h^j, a^{j+1}, \dots, a^n])$$

on développe par rapport à la $j^{\text{ème}}$ colonne on trouve :

$$det [a^1, a^2, \dots, a^{j-1}, h^j, a^{j+1}, \dots, a^n] = \sum_{i=1}^n h_i^j (cof A)_{ij} \Rightarrow \sum_{i,j=1}^n h_i^j (cof A)_{ij} = \langle cof A, H \rangle$$

donc $\nabla det A = cof A, \forall A \in \widetilde{\mathcal{M}}_n(\mathbb{R})$.

Soit $g : A \in \Omega \rightarrow g(A) = \log |\det A| \in \mathbb{R}$

g est différentiable (composée de deux fonctions différentiables)

$$\mathcal{D} g(A) : H \rightarrow \frac{D(det A)(H)}{det A} = \frac{\langle cof A, H \rangle}{det A} = \langle (A^{-1})^t, H \rangle$$

Donc $\nabla(\log|\det A|) = (A^{-1})^t, \forall A \in \Omega$ et si A est symétrique $\nabla(\log|\det A|) = A^{-1}$

III.3 : Présentation de la méthode.

Au problème (PSDP) on associe le problème barrière (PSDP) $_{\mu}$ suivant:

$$(PSDP)_{\mu} \begin{cases} \min_{X \in S^n} C \bullet X - \mu \ln(\det X) = f_{\mu}(X) \\ A_i \bullet X = b_i, \quad i = 1, \dots, m \\ X \succ 0 \end{cases}$$

Où $\mu > 0$ désigne le paramètre barrière.

La résolution de (PSDP) $_{\mu}$ est équivalente à celle de (PSDP) au sens que si $X^*(\mu)$ est une solution de (PSDP) $_{\mu}$ alors $X^* = \lim_{\mu \rightarrow 0} X^*(\mu)$ est une solution de (PSDP).

La fonction objectif $f_{\mu}(x)$ est suffisamment différentiable conformément au lemme précédent, de plus elle est strictement convexe et d'après les hypothèses (H_1), (H_2) et (H_3) le problème (PSDP) $_{\mu}$ admet pour chaque $\mu > 0$ une solution optimale unique $X^*(\mu)$ [39] vérifiant les conditions nécessaires et suffisantes d'optimalité de (KKT), représentées par le système suivant :

$$(8) \begin{cases} \sum_{i=1}^m y_i A_i + Z - C = 0 \\ A_i \bullet X - b_i = 0 \quad i = 1, \dots, m \\ XZ = \mu I \end{cases}$$

Le principe de cette méthode est de résoudre le système non linéaire (8) par la méthode de Newton, en partant d'un point intérieur non nécessairement réalisable ($X, Z \in S^n : X \succ 0, Z \succ 0$ et $y \in \mathbb{R}^m$), on obtient le

système linéaire suivant:

$$(9) \begin{cases} \sum_{i=1}^m \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^m y_i A_i - Z \\ A_i \bullet \Delta X = b_i - A_i \bullet X \quad i = 1, \dots, m \\ X(\Delta Z) + (\Delta X)Z = -XZ + \mu \sigma I, \quad 0 < \sigma < 1 \end{cases}$$

On peut réécrire le système (9) comme suit:

$$(10) \begin{cases} \sum_{i=1}^m \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^m y_i A_i - Z \\ A_i \bullet \Delta X = b_i - A_i \bullet X \quad i = 1, \dots, m \\ X\Delta ZZ^{-1} + \Delta X = -X + \mu \sigma Z^{-1}, \quad 0 < \sigma < 1 \end{cases}$$

on constater la symétrie de ΔZ à partir de la première équation du système (10), par contre la matrice ΔX n'est pas nécessairement symétrique du fait que le terme $X\Delta ZZ^{-1}$ n'est pas toujours symétrique. Cette difficulté est due à la structure du problème (SDP) et nécessite des efforts supplémentaires pour préserver la symétrie des itérés.

A ce propos, plusieurs procédures sont introduites, parmi lesquelles les plus célèbres alternatives suivantes : la direction de ALIZADEH, HAEBERLY et OVERTON (AHO), la direction de HELMBERG, KOJIMA et MONTEIRO (HKM) et la direction de Nesterov et Todd (NT). Il faut signaler que ces techniques font toujours l'objet de controverses concernant leurs conséquences parfois indésirables sur l'aspect algorithmique. Pour développer notre algorithme, nous avons choisi la technique (NT) à cause de sa compatibilité avec la théorie.

On remplace le terme $X\Delta ZZ^t$ par $P\Delta ZP^t$ tel que :

$$P = Z^{-1/2} (Z^{1/2} X Z^{1/2})^{1/2} Z^{-1/2}$$

donc le système (10) est remplacé par le système (11) suivant:

$$(11) \quad \begin{cases} \sum_{i=1}^m \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^m y_i A_i - Z \\ A_i \bullet \Delta X = b_i - A_i \bullet X \quad i = 1, \dots, m \\ P\Delta ZP^t + \Delta X = -X + \mu \sigma Z^{-1}, \quad 0 < \sigma < 1 \end{cases}$$

Maintenant ΔX est une matrice symétrique.

Le nouvel itéré est: $(X, y, Z) = (X, y, Z) + \alpha (\Delta X, \Delta y, \Delta Z)$

où $\alpha > 0$ est le pas de déplacement choisi de sorte que : $X \succ 0, Z \succ 0$

et φ décroît, où φ est la fonction de mérite associée au problème $(PSDP)_\mu$

définie par :

$$\varphi (X, y, Z) = X \bullet Z + r(X, y, Z)$$

$$\text{où } r(X, y, Z) = \| A' \bullet X - b \| + \| \sum_{i=1}^m y_i A_i + Z - C \|_F,$$

$$A' \bullet X = (A_1 \bullet X, A_2 \bullet X, \dots, A_m \bullet X)^t, b = (b_1, b_2, \dots, b_m)^t \text{ avec}$$

$$\|X\|_F = \sqrt{\text{tr}(XX^t)} = \sqrt{\text{tr}(X^2)} \text{ désigne la norme de Frobenius}$$

Si le test d'arrêt ($\varphi (X, y, Z) > \varepsilon$) n'est pas satisfait, on remplace μ par μ_l ($\mu_l < \mu$) et on réitère.

L'algorithme correspondant s'écrit:

Algorithme (NRSDP):

Début algorithme

Données : Soit $\varepsilon > 0$ un paramètre de précision fixé (suffisamment petit).

Initialisation: On choisit $(X, Z) > 0, y \in \mathbb{R}^m$ (arbitraires) et, $k = 0$

Tant que $\varphi > \varepsilon$ faire:

a) calculer $\mu = \frac{X \bullet Z}{n}$ et choisir $\sigma \in (0,1)$

b) calculer $(\Delta X, \Delta y, \Delta Z)$ solution du système linéaire (11):

c) on cherche $\alpha > 0$ tel que :

$X = X + \alpha \Delta X \succ 0, Z = Z + \alpha \Delta Z \succ 0$ et φ décroît

d) $X = X + \alpha \Delta X, y = y + \alpha \Delta y, Z = Z + \alpha \Delta Z$ et $\varphi = \varphi(X, y, Z)$

e) $K = K + 1$

Fin tant que

Stop dès que $\varphi \leq \varepsilon$

Fin algorithme.

III.4 : Convergence de l'algorithme.

Pour établir la convergence de cet algorithme nous avons besoin du théorème suivant :

Théorème 1 :

Si $0 < \alpha^k \leq 1$, la suite $\varphi^k = \varphi(X^k, y^k, Z^k)$ générée par l'algorithme (NRSDP) satisfait:

$$\varphi^{k+1} = (1 - \delta^k) \varphi^k$$

$$\text{Où } \delta^k = \frac{\alpha^k(1 - \sigma^k)X^k \bullet Z^k + \alpha^k v^k r^0 - (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k}{X^k \bullet Z^k + v^k r^0}$$

Preuve : La démonstration du théorème est basée sur le lemme suivant :

Lemme 3:

Soit $\{(X^k, y^k, Z^k)\}$ une suite générée par l'algorithme (NRSDP) alors:

$$1/ A_i \bullet X^{k+1} - b_i = v^{k+1} (A_i \bullet X^0 - b_i) \quad i=1, \dots, m$$

$$2/ \sum_{i=1}^m y_i^{k+1} A_i + Z^{k+1} - C = v^{k+1} (\sum_{i=1}^m y_i^0 A_i + Z^0 - C)$$

$$3/ X^{k+1} \bullet Z^{k+1} = (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k$$

$$\text{Où } v^{k+1} = (1 - \alpha^k) v^k = \prod_{j=0}^k (1 - \alpha^j) v^0, \quad v^0 = 1.$$

Preuve :

$$1/ A_i \bullet X^{k+1} - b_i = A_i \bullet (X^k + \alpha^k \Delta X^k) - b_i = (1 - \alpha^k) (A_i \bullet X^k - b_i) = (1 - \alpha^k) (1 - \alpha^{k-1}) (A_i \bullet X^{k-1} - b_i) =$$

:

$$= (1 - \alpha^k) (1 - \alpha^{k-1}) \dots (1 - \alpha^0) (A_i \bullet X^0 - b_i) = \prod_{j=0}^k (1 - \alpha^j) (A_i \bullet X^0 - b_i)$$

$$\begin{aligned}
 2/ \sum_{i=1}^m y_i^{k+1} A_i + Z^{k+1} - C &= \sum_{i=1}^m (y_i^k + \alpha^k \Delta y_i^k) A_i + (Z^k + \alpha^k \Delta Z^k) - C = \\
 (1 - \alpha^k) (\sum_{i=1}^m y_i^k A_i + Z^k - C) &= \dots = (1 - \alpha^k) (1 - \alpha^{k-1}) \dots (1 - \alpha^0) (\sum_{i=1}^m y_i^0 A_i + Z^0 - C) \\
 &= \prod_{j=0}^k (1 - \alpha^j) (\sum_{i=1}^m y_i^0 A_i + Z^0 - C)
 \end{aligned}$$

$$\begin{aligned}
 3/ X^{k+1} \bullet Z^{k+1} &= (X^k + \alpha^k \Delta X^k) \bullet (Z^k + \alpha^k \Delta Z^k) = \\
 X^k \bullet Z^k + \alpha^k (\Delta X^k \bullet Z^k + X^k \bullet \Delta Z^k) &+ (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k = \\
 X^k \bullet Z^k + \alpha^k (-X^k \bullet Z^k + n \mu^k \sigma^k) &+ (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k = \\
 X^k \bullet Z^k + \alpha^k (-X^k \bullet Z^k + \sigma^k) &+ (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k = \\
 (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k &+ (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k
 \end{aligned}$$

Démonstration du théorème :

$$\begin{aligned}
 \varphi^{k+1} &= \varphi (X^{k+1}, y^{k+1}, Z^{k+1}) \\
 &= X^{k+1} \bullet Z^{k+1} + r(X^{k+1}, y^{k+1}, Z^{k+1}) \\
 &= (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k + v^{k+1} r(X^0, y^0, Z^0) \\
 &= (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k + v^{k+1} r^0 \\
 &= (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k + v^k (1 - \alpha^k) r^0 \\
 &= (1 - \alpha^k + \sigma^k \alpha^k) X^k \bullet Z^k + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k + v^k r^0 - \alpha^k v^k r^0 \\
 &= X^k \bullet Z^k + v^k r^0 - \alpha^k X^k \bullet Z^k + \alpha^k \sigma^k X^k \bullet Z^k - \alpha^k v^k r^0 + (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k \\
 &= X^k \bullet Z^k + v^k r^0 \left(1 - \frac{\alpha^k (1 - \sigma^k) X^k \bullet Z^k + \alpha^k v^k r^0 - (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k}{X^k \bullet Z^k + v^k r^0} \right) \\
 &= \left(1 - \frac{\alpha^k (1 - \sigma^k) X^k \bullet Z^k + \alpha^k v^k r^0 - (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k}{X^k \bullet Z^k + v^k r^0} \right) \varphi^k \\
 &= (1 - \delta^k) \varphi^k
 \end{aligned}$$

$$\text{Où } \delta^k = \frac{\alpha^k(1-\sigma^k)X^k \bullet Z^k + \alpha^k v^k r^0 - (\alpha^k)^2 \Delta X^k \bullet \Delta Z^k}{X^k \bullet Z^k + v^k r^0}.$$

La fonction de mérite décroît d'une itération à l'autre d'un montant égal à $(1 - \delta^k)$, de plus on a le corollaire suivant :

Corollaire 1 :

La suite φ^k converge au moins linéairement et si δ^k tend vers 1 la convergence est superlinéaire. Rappelons que: $0 < \delta^k < 1$ si $0 < \alpha^k \leq 1$.

III.5 : Commentaire.

On peut conclure que ces méthodes constituent une solution convenable pour le problème d'initialisation. Reste à faire les aménagements nécessaires pour une mise en œuvre mettant en évidence les résultats établis.

CONCLUSION

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. Le seul inconvénient 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 et représente un handicap majeur pour ces méthodes originales et importantes.

Dans cette thèse, nous avons présenté des moyens qui apportent des vraies réponses aux questions théoriques, algorithmiques et numériques pour résoudre ce problème d'initialisation dans les méthodes de points intérieurs de type trajectoire centrale.

Notre étude est réalisée à travers trois classes de problèmes connus par leur intérêt théorique et pratique, à savoir, la programmation linéaire, la programmation quadratique convexe et la programmation semi définie.

Les résultats obtenus sont très encourageants et donnent lieu à d'autres perspectives dans le domaine de l'optimisation numérique.

RÉFÉRENCES

-
- [1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp.13-51.
- [2] F. Alizadeh, J. P. Haeberly, and M. Overton, *Primal-dual Interior –point Methods for Semidefinite Programming*, Technical Report 659, Computer Science Department, Courant Institute of Mathematical Sciences, New York University, New York, NY, 1994.
- [3] F. Alizadeh, J. P. Haeberly, and M. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746-768.
- [4] S. Bazara, H. D.Sherali, C. M. Shetty, *Nonlinear programming, theory and algorithm*, Second edition (1993).
- [5] J. F. Bonnans, J. C. Gilbert, C. Lemerechal, C. Sagastizabal, *Méthodes numérique d'optimisation*, INRIA.78153 Le Chesnay France (1995).
- [6] A. Coulibaly, *Méthodes de points intérieurs en programmation linéaire*, Thèse de l'université Blaise Pascal (1994).
- [7] J. C. Culioli, *Introduction à l'optimisation*, Edition Marketing (1994).
- [8] G. B. Dantzing, *linear programming and extensions*, Princeton University Press, Princeton, New York, (1963).
- [9] F. I. Dikin, Interactive solution of problem of linear and quadratique programming, Doklady Akademü Nauk SSR, Vol.174,PP.747-748,1967
- [10] A. V. Fiacco, G .P. McCormick, *Nonlinear programming sequential unconstrained minimization technique* (1968), J.Wiley, New York.
- [11] K. R. Frisch, *The logarithmic potential method for convex programming*, Unpublished manuscript, Institute of Economic, University of Oslo, Oslo, Norway (1955).

-
- [12] C. Ganzaga, *path-following methods for linear programming*. SIAM Review Vol 34.NO.2 (1992).
- [13] N. Karmarkar, *A new polynomial time algorithm for linear programming* Combinatorica, 4 (1984), 373-395.
- [14] A. Keraghel, *Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar*, Thèse de Doctorat Université Joseph Fourier Grenoble (1989).
- [15] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet, Mathematics Doklady, Vol 20, pp 191-194, 1979.
- [16] V. Klee and G. j. Minty, *How good is the simplex algorithm?* Inequalities, O. Shisha, ed, Academic Press, New York (1972), pp. 159-175.
- [17] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29-47.
- [18] M. Kojima, M. Shida, and S. Shindoh, *A Note on the Nesterov-Todd and the Kojima-Shindoh-Hara Search Directions in Semidefinite Programming*, Research Report B-313, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo, Japan, April 1996.
- [19] M. Kojima, M. Shida, and S. Shindoh, *A predictor-corrector interior-point algorithm for the smidefinite linear complementarity problem using the Alizadeh-Haeberly Overton search direction*, SIAM J. Optim., to appear.
- [20] N. Megiddo, *Pathways to the optimal set in linear programming*, Progress in Mathematical Programming interior-point (1989).

-
- [21] N. Megiddo, M. Kojima, S. Mizuno, *A primal-dual infeasible interior point algorithm for linear programming*, Mathematical Programming 61(1993) 263-280.
- [22] S. Mizuno, *Polynomiality of infeasible interior point algorithms for linear programming*, Mathematical Programming 67(1994) 109-119.
- [23] S. Mizuno, M. Kojima and A. Yoshise, *A primal-dual interior-point algorithm for linear programming*, Report NO.B188, TOKYO (1987).
- [24] S. Mizuno, M.J. Todd, and Y. Ye, *On adaptive step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 945-981.
- [25] Minoux, *Programmation mathématique théorie et algorithmes*, T.1 (1984), Dunod.Paris.
- [26] R.D.C. Monteiro, *Primal-dual path following algorithms for Semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663-678.
- [27] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part I: Linear programming*, Mathematical Programming, 44 (1989), pp. 27-41.
- [28] R.D.C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms: Part II Convex quadratic programming*, Mathematical Programming, 44 (1989), pp. 43-66.
- [29] Monteiro R. D. C, *Primal-dual path-following algorithms for semidefinite programming. SIAM Journal on Optimization*, 7, 1997, 663-678.
- [30] Monteiro R. D. C and Y. Zhang, *A unified analysis for a class of path-following primal-dual interior-point algorithms for Semidefinite programming*, Mathematical Programming, 81 (1998),pp. 281-299.

- [31] Y. E. Nesterov and A. S. Nemirovskii, *Polynomial barrier methods in convex programming*, Ekonomikai Matem. Metody, 24 (1988), pp. 1084-1091 (in Russian; English translation in Matekon: Translations of Russian and East European Math.Economics).
- [32] Y. E. Nesterov and A. S. Nemirovskii, *Self-concordant Functions and Polynomial Time Methods in Convex Programming*, preprint, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1989.
- [33] Y. E. Nesterov and A. S. Nemirovskii, *Optimization Over Positive Semidefinite Matrices: Mathematical Background and User's Manual*, Technical report, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1990.
- [34] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1994.
- [35] H. Roumili, *Etude qualitative des méthodes de points intérieurs non réalisables pour la programmation linéaire*, Thèse de Magister (1998), U.F.A Sétif- Algérie-.
- [36] H. Roumili, A. Keraghel and A. Yassine, *Infeasible primal-dual algorithm for minimizing convex quadratic problems*". Studia Univ. Babeş Bolyai, informatica, Volume XLIX, Number 2, 2004 pp. 81-90.
- [37] H. Roumili, A. Keraghel and A. Yassine, " *Infeasible Interior Point Method for Semidefinite Programs*" Applied Mathematical Sciences, Vol.1 2007, no.21, 1009-1018
- [38] L. Vandenberghe and S. Boyd, *A primal-dual potential reduction method for problems involving matrix inequalities*, Mathematical Programming, 69 (1995), pp. 205-236.
- [39] Wolkowicz, H. Saigal, R. and Vandenberghe L.: *Handbook of*

Semidefinite programming, Theory, Algorithms and Applications, Kluwer Academic Publishers, 2000.

[40] S.J.Wright, *Primal-dual interior point methods*, Copyright (1997) by SIAM.

[41] Y. Ye, *A class of projective transformations for linear programming*, SIAM J. Comput., 19(1990), pp. 457-466.

[42] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365-386.

[43] Y. hang, D.Zhang, *Superlinear convergence of infeasible interior point methods for linear programming*, Mathematical programming 66 (1994) pp. 361-377