UNIVERSITÉ ABDELHAMID IBN BADIS-MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES

Mémoire de fin d'étude
Pour l'obtention du diplôme de Master en Mathématiques
Cycle LMD
Spécialité : Modélisation Contrôle et Optimisation

Thème :

Méthode de Trajectoire Centrale Primal-Dual

Présenté par :
Melle BOUKHOBZA Meriem

Soutenu le 19/06/2013.

Les membres de jury

BELHAMETI	Omar	**Président**	MCA	**U. MOSTAGANEM.**
ABLAOUI	Houssin	**Examinateur**	MCA	**U. MOSTAGANEM.**
AMIR	ABED Samed	**Encadreur**	MCA	**U. MOSTAGANEM.**

Dédicace

" La joie et l'amertume sont les signes qui vont marquer certainement la fin de nos études au milieu des souvenirs inoubliables"

Je remercie DIEU qui a illuminé mon chemin "

Je dédie ce modeste travail à :

Ma chère mère de m'avoir donnée de la dignité et de courage et mon très cher père pour son soutien durant la période de mes études, que dieu les garde

Mes chers frères et Ma chère sœur.

A toute ma famille (BOUKHOBZA)

A mes cher(e)s ami(e) s

Ainsi qu'a mes ami(e) s de la promotion de 2ème année master (2012- 2013).

Je dédie aussi à tous ceux qui ont contribué de prés ou de loin dans l'élaboration de ce travail.

"Notre mère l'Algérie"

Remerciements

Au terme de ce travail je tiens à remercier : Monsieur AMIR ABDESSAMAD mon directeur de mémoire pour ses orientations qui m'ont été utiles pour la réalisation de ce travail. Monsieur BELHAMITI OMAR qui m'a fait l'honneur de présider le Jury. Monsieur ABLAOUI HOUSSIN pour avoir accepté de juger ce travail.

Mes vifs remerciements s'adressent également à : Tous les enseignants de département De mathematique pour leur enseignement.

Enfin, à tous ceux qui ont participé de prés ou de loin, à l'élaboration de ce mémoire, trouvent ici mes remerciements les plus sincères.

Merci

Table des matières

INTRODUCTION

Les méthodes de points intérieurs sont apparues dans les années cinquante. Ces méthodes reposent sur la fonction barrière logarithmique définit en 1955 par Frisch <cite>FRI</cite>. Cette fonction barrière logarithmique sera largement étudiée dans les années soixante, notamment dans le livre de Fiacco et McCormick <cite>FIA</cite>. C'est dans ce livre que le terme de points intérieurs sera introduit. Les itérés générés par les méthodes basées sur cette fonction sont strictement réalisables : ils restent à l'intérieur du domaine réalisable, d'où la terminologie de ces méthodes. L'engouement pour ces méthodes dans les années soixante est dû au développement de méthodes performantes en optimisation sans contrainte.
Avant 1984, tout problème d'optimisation linéaire se résolvait par la méthode du simplexe développé par Danzig <cite>DAN</cite> ou par une variante de celle ci. Des recherches ont été menées pour mettre au point une autre méthode mais aucune de celles proposées n'améliorait celle du simplexe. Aussi, pendant une quarantaine d'années, cette méthode domina l'optimisation linéaire.
Puis, dans les années 70, la théorie de la complexité devint une part intégrante de l'optimisation linéaire, si bien qu'on demanda aux méthodes développées de converger en un temps polynômial, c'est à dire de résoudre le problème en un nombre d'opérations qui doit être borné par un polynôme en la taille du problème. Mais la méthode du simplexe n'a pas cette propriété, comme l'ont montré Klee et Minty <cite>KLE</cite>.On se demanda alors si un algorithme d'optimisation linéaire avait cette propriété. En 1979, Khachian proposa un algorithme de programmation linéaire appelé méthode des ellipses de Khachian <cite>KHA</cite>. Bien que convergeant polynômialement en théorie, cet algorithme convergeait en pratique moins vite que le simplexe. Toutefois, Khachian montra théoriquement l'existence d'algorithmes à convergence polynômiale. Il restait maintenant à en trouver qui soient efficaces en pratique. Ce que fit Karmarkar en 1984 <cite>KAR</cite>. Il proposa en effet un algorithme de points intérieurs à convergence polynômiale pour résoudre des problèmes d'optimisation linéaire. Cela provoqua un regain d'intérêt pour les méthodes de points intérieurs, aussi bien en optimisation linéaire qu'en optimisation non linéaire.<cite>Phi</cite>.
Nous étudions dans ce mémoire, la méthode de trajectoire centrale primal-dual, qui reste l'une des méthodes de points intérieurs la plus utilisées dans les logiciels de programmation linéaire. Le premier chapitre est dédier à un exemple pathologique, sur lequel le simplexe donne de très mauvais résultats, sa compréhension exige les connaissances de base de la programmation linéaire. Ceci, nous a permit d'aborder le chapitre principale de ce mémoire qui est dédié entièrement à l'étude théorique de la méthode trajectoire centrale primal-dual. L'initialisation dans l'algorithme du simplexe se résume à trouver un sommet du polyèdre ou d'une manière équivalente une solution de base admissible, cette question est bien résolut et avec plusieurs approches. Cependant, le problème d'initialisation dans les méthodes de points intérieurs est complètement différent, vue qu'on démarre d'une solution strictement réalisable, ceci fera l'objet du troisième et dernier chapitre. En annex des tests numériques ont été réalisé.

Chapitre 1

Quand le simplexe échoue

L'algorithme du Simplexe peut ne pas prendre le plus court chemin pour atteindre l'optimum. Faire rentrer dans la base la variables de la plus grande marge réduite n'est pas nécessairement le meilleurs choix (même si c'est le plus naturel car c'est la variable qui accroît le plus la fonction objective). En 1970, Klee et Minty ont montré que, pour une classe de programmes linéaires la complexité de l'algorithme du Simplexe est exponentielle. Le nombre d'itérations qu'il faut pour que l'algorithme se termine est très grand que l'ordre de e^{N} . Ici N est une mesure de la taille du problème qui s'explicite à l'aide du nombre n de variables et de la longueur L du codage des données. Ce chapitre va se résumé à cet exemple historique, pour le lecteur non familiariser avec la méthode du simplexe nous le renvoyons à tout ouvrage de base sur la programmation linéaire en particulier <cite>FER</cite>. Klee et Minty, ont considéré le problème linéaire suivant écrit sous la forme standard, avec n un entier de $\mathbb{N}$.

$$\begin{cases} \max \sum_{j=1}^{n} 10^{n-j} x_j \\ 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-j}, \text{ pour } 1 \leq i \leq n \\ x_j \geq 0, \text{ pour } \ 1 \leq j \leq n \end{cases} \tag{1.0.1}$$

Exemple 1.0.1 *Prenons le cas où $n = 3$, le problème ci-dessus devient*

$$\begin{cases} \max 100x_1 + 10x_2 + x_3 \\ x_1 \leq 1 \\ 20x_1 + x_2 \leq 100 \\ 200x_1 + 20x_2 + x_3 \leq 10000 \\ x_j \geq 0, \textit{pour } j \ = \overline{1,3} \end{cases} \tag{1.0.2}$$

Le tableau simplexe initial est :

$$\begin{array}{cccccccc} x_1^{\downarrow} & x_2 & x_3 & s_1 & s_2 & s_3 & s.m & v.b \\ \boxed{1} & 0 & 0 & 1 & 0 & 0 & 1 & s_1 \longrightarrow \\ 20 & 1 & 0 & 0 & 1 & 0 & 100 & s_2 \\ 200 & 20 & 1 & 0 & 0 & 1 & 10000 & s_3 \\ 100 & 10 & 1 & 0 & 0 & 0 & 0 & z \end{array} \tag{1.0.3}$$

L'élément encadré dans le tableau ci-dessus est le pivot, la variable x_1 entre dans la base et s_1 en sort.

$$\begin{array}{cccccccc} x_1 & x_2^{\downarrow} & x_3 & s_1 & s_2 & s_3 & s.m & v.b \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & x_1 \\ 0 & \boxed{1} & 0 & -20 & 1 & 0 & 80 & s_2 \longrightarrow \\ 0 & 20 & 1 & -200 & 0 & 1 & 9800 & s_3 \\ 0 & 10 & 1 & -100 & 0 & 0 & 100 & z \end{array} \tag{1.0.4}$$

A la deuxième itération $x_2$entre et s_2 sort

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & s_1^{\downarrow} & s_2 & s_3 & s.m & v.b \\ 1 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & x_1 \longrightarrow \\ 0 & 1 & 0 & -20 & 1 & 0 & 80 & x_2 \\ 0 & 0 & 1 & 200 & -20 & 1 & 8200 & s_3 \\ 0 & 0 & 1 & 100 & -10 & 0 & 900 & z \end{array} \tag{1.0.5}$$

A la troisième, s_1 entre et x_1 sort

$$\begin{array}{cccccccc} x_1 & x_2 & x_3^{\downarrow} & s_1 & s_2 & s_3 & s.m & v.b \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & s_1 \\ 20 & 1 & 0 & 0 & 1 & 0 & 100 & x_2 \\ -200 & 0 & \boxed{1} & 0 & -20 & 1 & 8000 & s_3 \longrightarrow \\ -100 & 0 & 1 & 0 & -10 & 0 & -1000 & z \end{array} \tag{1.0.6}$$

A la quatrième, x_3 entre et s_3 sort

$$\begin{array}{cccccccc} x_1^{\downarrow} & x_2 & x_3 & s_1 & s_2 & s_3 & s.m & v.b \\ \boxed{1} & 0 & 0 & 1 & 0 & 0 & 1 & s_1 \longrightarrow \\ 20 & 1 & 0 & 0 & 1 & 0 & 100 & x_2 \\ -200 & 0 & 1 & 0 & -20 & 1 & 8000 & x_3 \\ 100 & 0 & 0 & 0 & 10 & -1 & -9000 & z \end{array} \tag{1.0.7}$$

A la cinquième, x_1 entre et s_1 sort

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & s_1 & s_2^{\downarrow} & s_3 & s.m & v.b \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & s_1 \\ 0 & 1 & 0 & -20 & \boxed{1} & 0 & 100 & x_2 \longrightarrow \\ 0 & 0 & 1 & 200 & -20 & 1 & 8000 & x_3 \\ 0 & 0 & 0 & -100 & 10 & -1 & -9100 & z \end{array} \tag{1.0.8}$$

A la sixième, s_2 entre et x_2 sort

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & s_1^{\downarrow} & s_2 & s_3 & s.m & v.b \\ 1 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & x_1 \longrightarrow \\ 0 & 1 & 0 & -20 & 1 & 0 & 80 & s_2 \\ 0 & 20 & 1 & -200 & 0 & 1 & 9800 & x_3 \\ 0 & -10 & 0 & 100 & 0 & -1 & -9900 & z \end{array} \tag{1.0.9}$$

A la septième, s_1 entre et x_1 sort

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & s_1 & s_2 & s_3 & s.m & v.b \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & s_1 \\ 20 & 1 & 0 & 0 & 1 & 0 & 100 & s_2 \\ 200 & 20 & 1 & 0 & 0 & 1 & 1000 & x_3 \\ -100 & -10 & 0 & 0 & 0 & -1 & -10000 & z \end{array} \tag{1.0.10}$$

La dernière ligne étant négative, l'algorithme s'arrête à la septième itération et donne la solution optimale $x^* = [1, 100, 1000]^\top$.

Klee et Menty, ont montré que pour tout $n \in \mathbb{N}$, leur exemple nécessite $2^n - 1$ itérations pour qui est convergence, d'où le caractère exponentiel de la méthode du simplexe. Malgré ceci, le simplexe reste l'algorithme le plus utilisé dans les logiciels commerciaux de la programmation linéaire. Dans le chapitre qui suit nous présenterons une méthode de point intérieur appelé trajectoire centrale primal-dual, tout en montrant son caractère polynomiale.

Chapitre 2

Méthode de trajectoire centrale Primal-Dual

L'idée générale des méthodes de trajectoire centrale [9] consiste à suivre un chemin centrale en prenant comme direction de déplacement celle de Newton . L'algorithme génère une suite de solutions réalisables intérieures (voir l'hypothèse ci-dessous) primal-dual $(x_n^\top, z_n^\top)$ convergent vers la solution optimale quand n tend vers l'infini.

2.1 Background Théorique

Considérons le programme linéaire écrit sous la forme standard :

$$(P)\begin{cases} \min c^\top x \\ Ax = b \\ x \geq 0 \end{cases}, \tag{2.1.1}$$

ainsi que son dual

$$(D)\begin{cases} \max b^\top y \\ A^\top y + z = c \\ z \geq 0 \end{cases} \tag{2.1.2}$$

où, A est une matrice de type $(m \times n)$, b et c sont des matrices uni-colonne de taille m et n respectivement. La méthode que nous désirons, est motivée par l'application de la méthode fonction de barrière logarithmique, elle consiste à étudier la famille des problèmes

$$(P_\mu)\begin{cases} \min c^\top x - \mu \sum_{j=1}^{n} \ln x_j \\ Ax = b \\ x \succ 0 \end{cases} \tag{2.1.3}$$

Où, ■0 est le paramètre de pénalité barrière. Cette technique est très utilisée dans la résolution des problèmes d'optimisation avec contraintes. On résous le problème pénalisé pour plusieurs valeurs du paramètre, en faisant tendre vers 0, nous obtenons une suite de point qui converge

en générale vers une solution du problème initial. Pour pouvoir utiliser cette technique, il est nécessaire de donner les trois hypothèses suivantes sur les deux problèmes (P) et (D).

Hypothèses

(a) L'ensemble

$$S = \{x \in \mathbb{R}^n,\, Ax = b,\, x > 0\} \text{ est non vide.} \tag{2.1.4}$$

(b) L'ensemble

$$T = \left\{(y, z) \in \mathbb{R}^m \times \mathbb{R}^n,\, A^\top y + z = c,\, z > 0\right\} \text{ est non vide.} \tag{2.1.5}$$

(c) Le rang de la matrice A est maximal, i.e., $rg(A) = m$.

Nous appelons un point appartenant à S et T, une solution réalisable intérieure des problèmes (2.1.1) et (2.1.2) respectivement. La nécessité de (a) est évidente pour l'admissibilité des contraintes du problème (2.1.3), nous verrons aussi celle de (b) et (c) dans les discussions qui suit. Notons par w le vecteur de $(\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n)$ tel que $w = (x, y, z)$. Une lettre minuscule x désignera le vecteur de $\mathbb{R}^n$, $x = (x_1, x_2, ..., x_n)^\top$, alors qu'une lettre majuscule X désignera la matrice diagonale dont les éléments diagonaux sont les composantes du vecteur x, on notera que $X = diag(x_1, x_2, ..., x_n)$. Nous noterons par

$$W = \{(x, y, z), x \in S, (y, z) \in T\} \tag{2.1.6}$$

On remarque que la fonction objective du problème (2.1.3) est strictement convexe, cela implique que le problème (2.1.3) possède au plus un minimum globale et que ce dernier, s'il existe est complètement caractérisé par les conditions de Karush, Kuhn et Tucker

$$\begin{cases} c - \mu X^{-1} e - A^\top y = 0 \\ Ax = b \\ x > 0 \end{cases}, \tag{2.1.7}$$

Où, e est le vecteur uni-colonne de $\mathbb{R}^n$ dont tous les éléments valent 1, alors que y est le multiplicateur de Lagrange associé aux contraintes d'égalité du problème (2.1.3).
En introduisant le z de $\mathbb{R}^n$, On aura

$$z = c - A^\top y, \tag{2.1.8}$$

en remplaçant (2.1.8) dans la première équation de (2.1.7), on obtient

$$Ze - \mu X^{-1} e = 0, \tag{2.1.9}$$

et en multipliant (2.1.9) par X on obtient

$$ZXe - \mu e = 0. \tag{2.1.10}$$

Le système (**??**) peut être réécrit de la manière équivalente suivante :

$$\begin{cases} ZXe - \mu e = 0 \\ Ax - b = 0, x > 0 \\ A^\top y + z - c = 0 \end{cases} \tag{2.1.11}$$

En désignant par $w(\mu) = (x(\mu), y(\mu), z(\mu))$ la solution du système non linéaire (2.1.11).
On a $w(\mu) \in W$.
Le saut de dualité au point $w \in W$ est par définition donné par,

$$g(w) = c^\top x - b^\top y. \tag{2.1.12}$$

En utilisant la troisième équation du système (2.1.11) on obtient,

$$g(w) = x^\top z \quad w \in W, \tag{2.1.13}$$

et la première équation de (2.1.11) nous donne,

$$g(w(\mu)) = n\mu. \tag{2.1.14}$$

Il est claire que $g(w(\mu))$ tend vers zéro quand μ tend vers zéro.

Proposition 2.1.1 ([?]) *Sous les hypothèses (2.1.4) et (2.1.5), si $\mu \to 0$, alors $(x(\mu))$ et $(y(\mu), z(\mu))$ convergent vers la solution optimale de (P) et (D) respectivement.*

Notons le vecteur

$$f(w) = ((f_1(w), ..., f_n(w))^\top \in \mathbb{R}^n \text{ avec } f_i(w) = x_i z_i, i = \overline{1, n}.$$

On déduit aussi que,

$$f_i(w(\mu)) = x_i(\mu) z_i(\mu) = \mu, i = \overline{1, ..., n}.$$

On note par Γ l'ensemble de solution $w(\mu)$ pour le système (2.1.11), i.e

$$\Gamma = \{w(\mu) = (x(\mu), y(\mu), z(\mu); \mu > 0\} \tag{2.1.15}$$

Γ est appelé trajectoire centrale associée au problème (2.1.3).

2.2 L'algorithme

Dans cette section, on note $w = (x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. L'algorithme génère une suite des points $w^k \in W$, $k = 0, 1, 2, ..., n$, avec w^0 un point initiale satisfaisant un critère de proximité par rapport à la trajectoire centrale, et $\widehat{w} = (\widehat{x}, \widehat{y}, \widehat{z})$ est la nouvelle itération tel que,

$$\begin{aligned} \widehat{x} &= x - \Delta x \\ \widehat{y} &= y - \Delta y \\ \widehat{z} &= z - \Delta z \end{aligned} \tag{2.2.1}$$

plus genèrale,

$$\widehat{w} = w - \Delta w, \text{ avec } \Delta w = (\Delta x, \Delta y, \Delta z). \tag{2.2.2}$$

Pour déterminer la direction Δw on utilise la direction de Newton associée au système (2.1.11). Si on note par $H(w) = H(x, y, z)$ la partie gauche du système (2.1.11), on aura d'après la méthode de Newton,

$$D_w H(w)\Delta w = H(w) \tag{2.2.3}$$

avec $D_w H$ le Jacobian de H, On a :

$$D_w H = \begin{bmatrix} z & 0 & x \\ A & 0 & 0 \\ 0 & A^\top & I \end{bmatrix}$$

et la direction de Newton est donnée par le système linéaire suivant,

$$Z\Delta x + X\Delta z = XZe - \widehat{\mu}e \tag{2.2.4}$$

$$A\Delta x = 0 \tag{2.2.5}$$

$$A^\top \Delta y + \Delta z = 0 \tag{2.2.6}$$

avec $\widehat{\mu} > 0$. Alors on peut trouver Δw d'après (2.2.4)

$$\Delta x = Z^{-1}\left[-X\Delta z + XZe - \widehat{\mu}e\right] \tag{2.2.7}$$

d'après(2.2.5),

$$A\Delta x = AZ^{-1}\left[-X\Delta z + XZe - \widehat{\mu}e\right] = 0$$

donc,

$$AZ^{-1}X\Delta z = AZ^{-1}(XZe - \widehat{\mu}e)$$

alors,

$$\Delta z = (AZ^{-1}X)^{-1}AZ^{-1}(XZe - \widehat{\mu}e)$$

qu'on peut écrire aussi comme suit

$$\Delta z = [A^\top (AZ^{-1}XA^\top)^{-1}AZ^{-1}](XZe - \widehat{\mu}e) \tag{2.2.8}$$

on remplace Δz dans (2.2.7), on obtient

$$\Delta x = \left[Z^{-1} - Z^{-1}XA^\top (AZ^{-1}XA^t)^{-1}AZ^{-1}\right](XZe - \widehat{\mu}e) \tag{2.2.9}$$

d'après(2.2.6),

$$\Delta y = -A^{\top^{-1}}\Delta z$$

donc,

$$\Delta y = -\left[(AZ^{-1}XA^\top)^{-1}AZ^{-1}\right](XZe - \widehat{\mu}e) \tag{2.2.10}$$

On peut à présent donner la description de l'algorithme, soient les deux valeurs θ et δ tel que :

$$0 \leq \theta \leq \frac{1}{2},\ 0 < \delta < \sqrt{n}, \tag{2.2.11}$$

et

$$\frac{\theta^2+\delta^2}{2(1-\theta)} \leq \theta(1-\frac{\delta}{\sqrt{n}}), \tag{2.2.12}$$

avec n, le nombre de colonne de la matrice A et

$$\left\|f(w^0)-\mu_0 e\right\| \leq \theta\mu_0 \tag{2.2.13}$$

où, la norme $\|.\|$ est la norme Euclidienne et $\mu_0 = \dfrac{(x^0)^\top z^0}{n}$.
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 (i,e, vérifie le critère de proximité).

Etapes de l'algorithme

Etape 1 On prend θ et δ fixé vérifiant (2.2.11) et (2.2.12) repectivement, $w^0 \in W$ vérifiant (2.2.13), la précision $\varepsilon > 0$ et on itialise $k = 0$.

Etape 2 Si $\left(x^k\right)^\top z^k \leq \varepsilon$, Stop.

Etape 3 Sinon on calcul $\mu_{k+1} = \mu_k(1-\frac{\delta}{\sqrt{n}})$ et $\Delta w^k = \Delta w(w^k, \mu^{k+1})$.

Etape 4 Calcul de $w^{k+1} = w^k - \Delta w^k$, on pose $k = k+1$ et aller en 2.

2.3 Convergence

Si le point courant (x, z) vérifie le critère de proximité par rapport à une valeur μ donnée, et si on choisit une valeur convenable du paramètre $\widehat{\mu} > 0$, alors le nouveau point $(\widehat{x}, \widehat{z})$ reste voisin de la trajectoire centrale, plus précisément on a le théorème suivant.

Théorème 2.3.1 *Soient θ et δ des constantes satisfaisant la relation (2.2.11). Supposons que $w = (x, y, z)$ est tel que*

$$\|f(w)-\mu e\| \leq \theta\mu \tag{2.3.1}$$

où, $\mu = \dfrac{x^\top z}{n}$, et soit $\widehat{\mu} > 0$ définit

$$\widehat{\mu} = \mu(1-\frac{\delta}{\sqrt{n}}) \tag{2.3.2}$$

considérons le point $\widetilde{w} = (\widetilde{x}, \widetilde{y}, \widetilde{z}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ avec,

$$\widehat{w} = w - \Delta w \tag{2.3.3}$$

où,

$$\Delta w = \Delta w(w, \widehat{\mu}). \tag{2.3.4}$$

Alors,

$$\|f(\widehat{w})-\widehat{\mu} e\| \leq \theta\widehat{\mu} \tag{2.3.5}$$

$$\widehat{w} \in W \tag{2.3.6}$$

$$g(\widehat{w}) = \widehat{x}^\top \widehat{z} = n\widehat{\mu}. \tag{2.3.7}$$

La preuve de ce théorème sera donnée à la fin de cette section.

Corollaire 2.3.1 *La suite de points (w^k) engendrée par l'algorithme satisfait*

1. $\left\|f(w^k) - \mu_k e\right\| \leq \theta\mu_k$, *pour tout* $k = \overline{1,n}$.
2. $w^k \in W$, *pour tout* $k = \overline{1,n}$.
3. $g(w^k) = x^{k^t} z^k = n\mu_k$, *pour tout* $k = \overline{1,n}$. *Où,*

$$\mu_k = \mu_0(1 - \frac{\delta}{\sqrt{n}})^k,\ k = \overline{1,n} \tag{2.3.8}$$

Proposition 2.3.1 *Le nombre total d'itérations effectuées par l'algorithme ne dépasse pas* $\widehat{k} = \left[\ln\left(n\varepsilon^{-1}\mu_0\right)\frac{\sqrt{n}}{\delta}\right]$ *où,* $\varepsilon > 0$ *désigne la tolérance du saut de dualité et* $\mu_0 = \frac{x^{0^t} z^0}{n}$ *est le paramètre de pénalité initiale.*

Preuve. L'algorithme prend fin lorsque $n\mu_k \leq \varepsilon$. Ainsi, il suffit de montrer qu'à l'itération $\widehat{k}$ le critère d'arrêt est vérifié. Par la définition de $\widehat{k}$, nous avons

$$\ln\varepsilon \geq -\frac{\widehat{k}\delta}{\sqrt{n}} + \ln(n\mu_0)$$

donc,

$$\ln\varepsilon \geq \widehat{k}\ln(1 - \frac{\delta}{\sqrt{n}}) + \ln(n\mu_0)$$

d'où,

$$\ln\varepsilon \geq \ln\left[n\mu_0(1 - \frac{\delta}{\sqrt{n}})^{\widehat{k}}\right]$$

il suit que,

$$\ln\varepsilon \geq \ln n\mu_{\widehat{k}}.$$

□

Proposition 2.3.2 *Soit* w, $\widehat{\mu}$, Δw *et* $\widehat{w}$ *comme ci-dessus. Alors,*

$$f_i(\widehat{w}) = \widehat{\mu} + \Delta x_i \Delta z_i \tag{2.3.9}$$

$$(\Delta x)^\top(\Delta z) = 0 \tag{2.3.10}$$

$$g(\widehat{w}) = n\widehat{\mu} \tag{2.3.11}$$

Preuve.

1. Par definition on a pour tout $i = \overline{1,n}$

$$\begin{aligned} f_i(\widehat{w}) &= \widehat{x}_i\widehat{z}_i \\ &= (x_i - \Delta x_i)(z_i - \Delta z_i) \\ &= x_i z_i - (x_i\Delta z_i + z_i\Delta x_i) + \Delta x_i \Delta z_i \end{aligned}$$

d'après la relation (2.2.4) on trouve,

$$f_i(\widehat{w}) = \widehat{\mu} + \Delta x_i \Delta z_i$$

2. On multiplie (2.2.5) par $\Delta y^\top$ et (2.2.6) par $\Delta x^\top$ on obtient,

$$\begin{aligned} \Delta y^\top A \Delta x &= 0 \\ (\Delta x)^\top A^\top \Delta y + (\Delta x)^\top \Delta z &= 0 \end{aligned}$$

donc on trouve,

$$(\Delta x)^\top \Delta z = 0$$

3. On a

$$\begin{aligned} g(\widehat{w}) &= \sum_{i=1}^{n} f_i(\widehat{w}) \\ &= \sum_{i=1}^{n} \widehat{\mu} + \Delta x_i \Delta z_i \end{aligned}$$

d'après(2.3.1) on trouve,

$$g(\widehat{w}) = \sum_{i=1}^{n} \widehat{\mu} = n\widehat{\mu}$$

□

Lemme 2.3.1 *Soit r, s et t des vecteurs réels, tel que $r + s = t$ et $r^\top s \geq 0$. Alors,*

$$RSe \leq \frac{\|t\|^2}{2} \tag{2.3.12}$$

où, R et S sont des matrices diagonales correspondent aux vecteurs r et s respectivement.

Preuve. En utilisant les hypothèses du lemme nous obtenons,

$$\begin{aligned} \|r\|^2 + \|s\|^2 &\leq \|r\|^2 + \|s\|^2 + 2r^\top s \\ \|r\|^2 + \|s\|^2 + 2r^\top s &= \|r+s\|^2 = \|t\|^2 \end{aligned}$$

donc,

$$\frac{\|t\|^2}{2} \geq \|r\|\,\|s\| \geq \|RSe\| \tag{2.3.13}$$

□

Lemme 2.3.2 *Soit Δf, définit comme précédemment ($\Delta f = (\Delta X)(\Delta Z)$). Alors,*

$$\|\Delta f\| \leq \frac{\|f(w) - \widehat{\mu}e\|^2}{f_{\min}} \tag{2.3.14}$$

où

$$f_{\min} = \min\{x_i z_i; i = 1, ..., n\}$$

Preuve. Soit $D = (Z^{-1}X)^{\frac{1}{2}}$. Multiplions les deux côtés de l'équation (2.2.4) par $(XZ)^{\frac{-1}{2}}$. On obtient

$$D^{-1}\Delta x + D\Delta z = (XZ)^{\frac{-1}{2}}(f(w) - \widehat{\mu}e),$$

d'après (2.3.10) nous avons,

$$(D^{-1}\Delta x)^\top (D\Delta z) = 0$$

en raison de ces deux relations, nous pouvons appliquer le lemme avec $r = D^{-1}\Delta x, s = D\Delta z$ et $t = (XZ)^{\frac{-1}{2}}(f(w) - \widehat{\mu}e)$. On obtient,

$$\begin{aligned}
\|\Delta f\| &= \|(\Delta X)(\Delta Z)e\| \\
&= \left\|(D^{-1}\Delta x)(D\Delta z)e\right\| \\
&\leq \frac{1}{2}\left\|(XZ)^{\frac{-1}{2}}(f(w) - \widehat{\mu}e)\right\|^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}\frac{(f_i(w) - \widehat{\mu})^2}{x_i z_i} \\
&\leq \frac{\|f(w) - \widehat{\mu}e\|}{2f_{\min}}
\end{aligned}$$

□

Démontrons à présent le théorème.

Preuve. D'après la relation (2.3.9), pour montrer (2.3.5) il faux montrer que

$$\|\Delta f\| \leq \theta\widehat{\mu}$$

et d'après (2.3.14) on montre que

$$\frac{\|f(w) - \widehat{\mu}e\|}{2f_{\min}} \leq \theta\widehat{\mu}.$$

On a

$$\mu = \frac{x^\top z}{n}, \tag{2.3.15}$$

et

$$(f(w) - \mu e)^\top e = 0, \tag{2.3.16}$$

$$\|e\| = \sqrt{n}. \tag{2.3.17}$$

Donc d'après (2.3.15) ,(2.3.16), (2.3.17) et (2.3.1), (2.3.2).On obtient

$$\begin{aligned}\|f(w)-\widehat{\mu}e\|^2 &= \|f(w)-\mu e\|^2+\|\mu e-\widehat{\mu}e\|^2\\ &\leq (\theta\mu)^2+(|\mu-\widehat{\mu}|\,\|e\|)^2\\ &= (\theta\mu)^2+(\delta\mu)^2\\ &= (\theta^2+\delta^2)\mu^2\end{aligned}$$

D'après (2.3.1) on a

$$\|f_{\min}-\mu e\|\leq\theta\mu$$

Donc,

$$f_{\min}-\mu\geq-\theta\mu$$

Alors,

$$f_{\min}\geq(1-\theta)\mu$$

Nous avons

$$\frac{\|f(w)-\widehat{\mu}e\|}{2f_{\min}}\leq\frac{\theta^2+\delta^2}{2(1-\theta)}\mu,$$

et d'après (2.2.12) et (2.3.2). On a

$$\frac{\|f(w)-\widehat{\mu}e\|}{2f_{\min}}\leq\theta(1-\frac{\delta}{\sqrt{n}})\mu=\theta\widehat{\mu}$$

Montrons (2.3.6) à partir de (2.2.5) et (2.2.6) et $\widehat{w}=(\widehat{x},\widehat{y},\widehat{z})$ satisfait

$$A\widehat{x}=b$$

et

$$A^\top\widehat{x}+\widehat{z}=c$$

nous avons juste à montrer que $\widehat{x}$ et $\widehat{z}$ des vecteurs strictement positifs et on conclu que $\widehat{w}\in W$. On démontre par absurde. Si

$$\widehat{x}_i<0 \text{ où } \widehat{z}_i<0.$$

D'après (2.2.1) on a

$$\Delta x_i\Delta z_i>x_iz_i$$

Et pour un indice i, d'après (2.3.5), nous avons

$$\widehat{x}_i\widehat{z}_i\geq(1-\theta)\widehat{\mu}>0,$$

donc il faut que

$$\widehat{x}_i<0\text{et }\widehat{z}_i<0,$$

alors,

$$\Delta x_i\Delta z_i>x_iz_i.$$

D'autre part, on a,

$$\Delta x_i\Delta z_i\leq\|\Delta f\|\leq\theta\widehat{\mu}\leq\theta\mu,$$

et on a,

$$\theta\mu>x_iz_i\geq(1-\theta)\mu$$

Contradiction avec $\theta<\frac{1}{2}$.
Finalement signalons que la preuve de (2.3.7) se déduit de la relation (2.3.11). □

Chapitre 3

Initialisation

Le choix du point de départ est un problème pratique important avec effet significatif sur la fiabilité de l'algorithme .Un mouvais choix (x^0, y^0, z^0) satisfaire uniquement les conditions minimales $x^0 > 0$ et $z^0 > 0$ entrainent souvent des problème de divergence . Nous décrivons ici une méthode heuristique qui trouve un point de départ qui satisfait assez bien les contraintes d'égalité dans les problèmes primal et dual, tout en conservant la positivité des composantes x et z, et en évitant de trop grandes valeur de ces composants [12].
Tout d'abord, nous trouvons un vecteur $\widetilde{x}$ de norme minimale respectant la contrainte primal

$$Ax = b, \tag{3.0.1}$$

et $(\widetilde{y}, \widetilde{z})$ un vecteur satisfaire la contrainte dual,

$$A^\top y + z = c, \tag{3.0.2}$$

telle que $\widetilde{z}$ est de norme minimale.
Autrement dit, nous résolvons les problèmes suivants :

$$\begin{cases} \min \frac{1}{2} x^\top x \\ Ax = b \end{cases}, \tag{3.0.3}$$

et

$$\begin{cases} \min_{(y,z)} \frac{1}{2} z^\top z \\ A^\top y + z = c \end{cases}. \tag{3.0.4}$$

Pour résoudre ces problèmes, nous utilisons le théorème de (K-K-T).
On a le lagrangien des problèmes (3.0.3) et (3.0.4) :

$$l(x, \lambda) = \frac{1}{2} x^\top x + \lambda^\top (Ax - b), \tag{3.0.5}$$

et

$$l(y, z, \mu) = \frac{1}{2} z^\top z + \mu^\top (A^\top y + z - c). \tag{3.0.6}$$

On applique les conditions de (K-K-T), on trouve

$$\begin{cases} \nabla l(x,\lambda) = x + \lambda^\top A = 0 \\ Ax - b = 0 \end{cases}, \tag{3.0.7}$$

et

$$\begin{cases} \nabla l(y,z,\mu) = \begin{pmatrix} z \\ 0 \end{pmatrix} + \mu^\top \begin{pmatrix} 1 \\ A^\top \end{pmatrix} = 0 \\ A^\top y + z - c = 0 \end{cases}.$$

D'après (3.0.7) on a

$$x = A^\top \lambda, \tag{3.0.8}$$

on remplace dans (3.0.1), on trouve,

$$AA^\top \lambda = b \tag{3.0.9}$$

donc,

$$\lambda = (AA^\top)^{-1} b \tag{3.0.10}$$

on remplace dans (3.0.8), on trouve,

$$x = A^\top (AA^\top)^{-1} b. \tag{3.0.11}$$

Et d'après (**??**) on a,

$$z = \mu^\top \tag{3.0.12}$$

et,

$$z = c - A^\top y \tag{3.0.13}$$

on remplace(3.0.12) dans (3.0.2) on trouve :

$$A^\top y + \mu^\top - c = 0$$

On multplie par A. On trouve,

$$AA^\top y + A\mu^\top - Ac = 0$$

donc, d'après (**??**) on trouve,

$$AA^\top y - Ac = 0,$$

et

$$y = (AA^\top)^{-1} Ac, \tag{3.0.14}$$

avec λet μ sont des paramètres de complémentarité.
Alors d'aprés (3.0.11) et (3.0.13) et (3.0.14) les solutions de (3.0.3) et (3.0.4) respectivement sont :

$$\begin{aligned} \widetilde{x} &= A^\top (AA^\top)^{-1} b \\ \widetilde{y} &= (AA^\top)^{-1} Ac \\ \widetilde{z} &= c - A^\top \widetilde{y} \end{aligned}$$

En générale x et z admettent des composantes négatives, donc ils ne sont pas appropriées. Pour être utilisées comme un point de départ, nous introduisons les variables

$$\begin{aligned} \delta_x &= \max(-\frac{3}{2}\min_i \widetilde{x}_i, 0) \\ \delta_z &= \max(-\frac{3}{2}\min_i \widetilde{z}_i, 0) \end{aligned}$$

et réglons les vecteurs x et z comme suit,

$$\begin{aligned} \widehat{x} &= \widetilde{x} + \delta_x e \\ \widehat{z} &= \widetilde{z} + \delta_z e \end{aligned}$$

qui sont solutions des problèmes (3.0.3) et (3.0.4), respectivement, avec $e = (1, 1, ., ...1)^\top$. De toute évidence, nous avons $\widehat{x} > 0$ et $\widehat{z} > 0$. Pour s'assurer que les composantes de x^0 et z^0 ne sont pas trop proches de zéro et pas trop distant, nous ajoutons deux scalaires définies comme suit,

$$\begin{aligned} \widehat{\delta_x} &= \frac{1}{2}\frac{\widehat{x}^\top \widehat{z}}{e^\top \widehat{z}}, \\ \widehat{\delta_z} &= \frac{1}{2}\frac{\widehat{x}^\top \widehat{z}}{e^\top \widehat{x}} \end{aligned}$$

A noter que $\widehat{\delta_x}$ est la taille moyenne des composantes de $\widehat{x}$, pondérée par les composantes correspondantes de $\widehat{z}$, de même pour $\widehat{\delta_z}$.
Enfin , nous définissons le point de départ comme suit,

$$\begin{aligned} x^0 &= \widehat{x} + \widehat{\delta_x} e \\ y^0 &= \widetilde{y} \\ z^0 &= \widehat{z} + \widehat{\delta_z} e. \end{aligned}$$

Le coût numérique de trouver (x^0, y^0, z^0), par ce schéma est pratiquement le même comme une étape de la méthode primal-dual.
Dans certains cas, nous avons une connaissance à priori sur la solution, peut être dans la forme d'une solution à un programme linéaire similaire.

CONCLUSION

La méthode de trajection centrale primal-dual est donc performant théoriquement et numériquement, comme le montrent les résultats numériques présentés ci-dessous. Elle est robuste car c'est une méthode simple : à chaque itération, on résout un système linéaire et on calcule un pas. Il y a peu de paramètres à régler. Le celle inconvénient de cette méthode est le recherche de point initiale.

Bibliographie

[1] Andreas A Wu-Sheng Lu,Practical Optimization : Algorithms and Engineering Applications,Department of Electrical and Computer Engineering University of Victoria, Canada ,2007

[2] G. B. DANTZIG, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

[3] M. C. Ferris, O. L. Mangasarian and S. J. Wright. Linear Programming With MATLAB, Society for Industrial and Applied Mathematics Philadelphia, 2007.

[4] A.V.FIACCO ETG.P.McCORMICK,Nonlinear programming :Sequential unconstrained minimization techenique,John Wiley and Sons, Inc., New York-London-Sydney, 1968.

[5] K. R. FRISCH, The logarithmic potential method of convex programming, , University Institute of Economics, Oslo, Norway.Memorandum of May 13 (1955).

[6] N. KARMARKAR, A new polynomial-time algorithm for linear programming, Combinatorics 4 (1984), pp. 373-395.

[7] L. G. KHACHIAN, A polynomial algorithm in linear programming, Soviet Mathematics Doklady, 20 (1979), pp. 191-194.

[8] V. KLEE ET G. J. MINTY, How good is the simplex algorithm ?, Inequalities, O. Shisha, ed, Academic Press, New York (1972), pp. 159-175.

[9] D.C. Monteiro and I. Adler. Interior Path Following Primal-Dual Algorithms. Part I ; Linear Programmaing, Mathematical Programming 44 (1989) 27-41.

[10] Philippe S,Thèse Méthodes de points intérieurs et de quasi-Newton,Universite de limoges U.F.R. de sciences et techniques,2002

[11] P.Pesneau Pierre.pesneau@math.V_bordeaux1.fr,Programmation linéaire cours1,Université .Bordeaux1 Bàt A33_Bur 265

[12] Thomas V. M ,Sid I. R ,Ste M. Robinson,Numerical Optimization,Springer Series in Operations Research and Financial Engineering

Chapitre 4

ANNEX

Sous programme pour calculer les points initiale $[x_0, y_0, z_0]$

(i) function [v]=initial(A,b,c,n)

Proposition 4.0.3 –

– *xt=A'*inv(A*A')*b ;*
*yt=inv(A*A')*A*c ;*
*zt=c-A'*yt ;*
*mx=-3/2*min(xt) ;*
*mz=-3/2*min(zt) ;*
gamax=max(mx,0) ;
gamaz=max(mz,0) ;
*xs=xt+gamax*ones(n,1) ;*
*zs=zt+gamaz*ones(n,1) ;*
*xsh=xs'*zs ;*
*ze=ones(n,1)'*zs ;*
*xe=ones(n,1)'*xs ;*
*gamaxh=0.5*xsh/ze ;*
*gamazh=0.5*xsh/xe ;*
*x0=xs+gamaxh*ones(n,1) ;*
y0=yt ;
*z0=zs+gamazh*ones(n,1) ;*
v=[x0 ; y0 ; z0] ;

Le programme principale pour calculer la solution optimal primal on utilise la méthode du point interieur

(ii) clear all

Proposition 4.0.4 –

```
– A=[1 1 1];
  b= 1;% le seconde membre
  c=[-2 1 -3]';%Vecteur de la fct objectif
  eps=10e-6;
  n=length(c);
  m=length(b);
  p=7*sqrt(n);
  v=initial(A,b,c,n);%le vecteur initiale [x0 y0 z0]
  h=2*n+m;
  x0=v(1 :n);%vecteur initiale de problème primal
  z0=v(n+m+1 :h);%l'un du vecteur initiale de problème dual
  X=diag(x0);%la matrice qui contien x0 dans le diagonale
  Z=diag(z0);%la matrice qui contien z0 dans le diagonale
  w0=v;
  k=0;%k c'est le nbr des itérations
  while x0'*z0 >eps %boucle pour calculè le neuvaux variable x ,z
      mu1=z0'*x0/(n+p);% est la valeur de logarithemique barier
           X=diag(x0);
           Z=diag(z0);
          y=x0-mu1*inv(Z)*ones(n,1);
              D=inv(Z)*X;
               U = inv(A*D*A');
                dy0 = U*A*y;
            dz0 = -A'*dy0;
             dx0=-y-D*dz0;
            dw0 = [dx0;dy0;dz0]%la direction choisie pour générer la prochaine
               %itération
              w1 = w0+dw0;% le nouvaux vecteur
            x0=[w1(1 :n)];
                 y0=w1(n+m);
                z0=[w1(n+m+1 :2*n+m)];
                    w0 = w1;
                    k=k+1;
  end
  wopt=w0%la solution primal-dual optimal
  xopt=x0%la solution primal optimal
  zopt=z0%la solution dual optimal
```

Exemple 4.0.1 *On considère le problème suivent :*

$$\begin{cases} \min f(x) = -2x_1 + x_2 - 3x_3 \\ x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

:avec

$$\begin{aligned} c &= [-2, 1, -3]^\top \\ A &= [1, 1, 1] \\ b &= 1 \end{aligned}$$

si on utilise le programme en matlab ,l'algorithme converge après neuf itérations à la solution

$$xopt = \begin{bmatrix} 0.000000 \\ 0.000000 \\ 1.000000 \end{bmatrix}$$