

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Université 8 Mai 1945 - Guelma

Faculté des Mathématiques et de l'Informatique
et des Sciences de la Matière
Département de Mathématiques



Mémoire

Présenté en vue de l'obtention du diplôme de

Master en Mathématiques

Option : EDP Et Analyse numérique

Par :

Belkharchiche Imane

Intitulé

**Méthode de trajectoires centrale pour résoudre un
programme convexe.**

Dirigé par : Bouafia Mousaab

Devant le jury

PRESIDENT
RAPPORTEUR
EXAMINATEUR

Bouhadjar Slimane. Docteur
Bouafia Mousaab. Docteur
Bahloul Tarek. Docteur

Univ-Guelma
Univ-Guelma
Univ Guelma

Session Juin 2020

Dédicace

Je dédie ce modeste travail avec un profond amour à celle qui m'a arrosé de tendresse et d'espoirs, à la source d'amour incessible qui m'a bénie par ses prières, ma mère.

À mon très cher père, pour ses encouragements, son soutien, surtout pour son amour et son sacrifice afin que rien n'entrave le déroulement de mes études et à qui je dois tout l'amour et le respect.

*Au petit rayon de soleil qui éclaire ma vie, à mon petit ange, ma fille
Éléna.*

À toute ma famille qui n'a jamais cessé de procurer l'aide nécessaire pour réaliser ce mémoire.

Remerciement

Après avoir rendu grâce à Dieu le tout Puissant et le Miséricordieux, je tiens à remercier vivement tous ceux qui, de près ou de loin ont participé à la rédaction de ce document. Il s'agit plus particulièrement de :

- Monsieur Bouafia Mousaab docteur à l'université du 08 mai 1945, pour sa disponibilité, sa rigueur scientifique et son sens d'écoute et d'échange.
- Monsieur Tarek Chiheb, le chef de département de la faculté des mathématiques à l'université du 08 mai 1945 – Guelma.

Je désire remercier également :

- Bouhadjar Slimane, Docteur à l'université du 08 mai 1945– Guelma,
- Bahloul Tarek, Docteur à l'université du 08 mai 1945– Guelma,

d'avoir accepté de juger ce travail et d'en être l'examineur.

Je remercie mes très chers parents, qui ont toujours été là pour moi pour leurs encouragements, leurs sentiments et leurs prières pour moi. Je remercie mes frères et sœurs surtout ma sœur Hanane pour son soutien inconditionnel et ses encouragements.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

المخلص:

في هذه المذكرة تطرقنا الى دراسة نظرية وخوارزمية لطريقة المسار المركزي لحل مسائل الأمثلة للبرامج الخطية بالطريقة العادية والطريقة التقليدية ثم نوسع الدراسة للبرامج المحدبة حيث نبحت عن القيمة الصغرى المقربة لدالة محدبة على مجموعة قيود خطية بواسطة هذه الطريقة.

الكلمات المفتاحية:

طريقة المسار المركزي - الدالة محدبة - البرامج الخطية - البرامج المحدب

Résumé :

Dans ce mémoire, d'une part, on entame une étude théorique et algorithmique de la méthode de trajectoire centrale pour résoudre les problèmes des exemples des programmes linéaires avec la méthode de la trajectoire centrale classique et la méthode de la trajectoire centrale avec poids. Et d'autre part, on élargit l'étude des programmes convexes où on cherche la valeur minimisée approximative d'une fonction convexe sur l'ensemble des contraintes linéaires avec cette méthode.

Mots clés :

La méthode de trajectoire centrale, La fonction convexe, Les programmes linéaires, Les programmes convexes.

Abstract:

In this thesis, we start a theoretical and algorithmic study of the central trajectory method to solve the problems of the examples of linear programs with the classical central trajectory method and the central trajectory method with weight, then we expand the study of convex programs where we seek the approximate minimized value of a convex function on the set of linear constraints with this method.

Key words :

The central trajectory method, The convex function, The linear programs, The convex programs.

Table des matières

| | |
|--|-----------|
| Introduction générale | 3 |
| 1 Outils de base | 5 |
| 1.1 Programmation mathématique | 8 |
| 1.2 Programmation linéaire | 10 |
| 2 Méthode de Trajectoire Centrale pour la programmation linéaire | 13 |
| 2.1 Présentation de la méthode classique | 13 |
| 2.1.1 Concept de proximité | 16 |
| 2.2 Méthode de Trajectoire Centrale avec poids pour la programmation linéaire | 17 |
| 2.2.1 Présentation de la méthode | 17 |
| 2.2.2 Etude de la convergence | 20 |
| 3 Présentation et principe de la méthode pour minimiser une fonction convexe non linéaire | 21 |
| 3.1 Méthode de la trajectoire centrale classique | 21 |
| 3.1.1 Description de la méthode | 21 |
| 3.2 Méthode de trajectoire centrale avec poids | 25 |
| 3.2.1 Présentation de la méthode : | 25 |
| 3.2.2 Calcul de la direction de descente (dx, dy, ds) | 26 |
| 3.2.3 Conclusion | 27 |

Introduction générale

Dans ce travail, on va élargir l'étude des problèmes de minimisation du fonction convexe sur les contraintes linéaires avec la méthode de trajectoire centrale .

Les méthodes de trajectoire centrale (TC) ont été introduites à la même époque que les méthodes de point intérieur et pleinement développées au début des années 90.

Elles possèdent de bonnes propriétés théoriques : une complexité polynomiale et une convergence linéaire. Les algorithmes de trajectoire centrale restreintes les itérés à un voisinage de la trajectoire centrale. Ce voisinage est une courbe de points strictement réalisables.

Les méthodes primales-duales de points intérieurs IPMs rentrent dans le cadre du chemin central et sont les méthodes les plus efficaces du point de vue pratique, en particulier, pour les problèmes de grande taille. Actuellement les chercheurs s'intéressent à l'étude de l'amélioration du comportement de l'algorithme en regardant en particulier la complexité algorithmique de méthodes de trajectoire centrale via une fonction noyau (etude prespective)

le mémoire est composé de trois chapitres. Dans le premier, on présente des notions de base sur l'analyse convexe , la programmation mathématique et la programmation linéaire . Le second chapitre expose la méthode de trajectoire centrale pour la programmation linéaire . Dans ce chapitre, nous proposons d'une part la présentation de la méthode classique et le Concept de proximité et d'autre part , la présentation de la méthode de trajectoire centrale avec poids pour la programmation linéaire et l'étude de

la convergence.

Enfin, le troisième chapitre est consacré à la présentation et principe de la méthode pour minimiser une fonction convexe non linéaire ce dernier comprend la méthode de la trajectoire centrale classique et la méthode de trajectoire centrale avec poids.

Chapitre 1

Outils de base

Dans cette partie, on présente un rappel des notions fondamentales de l'analyse convexe et de la programmation mathématique qui serviront d'appuis pour la suite. Pour plus de détails voir [8, 14].

a) Ensembles convexes :

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

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

- C est dit affine si :

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

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

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

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

C peut s'écrire sous la forme matricielle suivante :

$$C = \{x \in \mathbb{R}^n / Ax \leq b\}.$$

- S_n est un (n -simplexe) s'il est de la forme :

$$S_n = \left\{ x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1 \right\}.$$

- Un point $x \in S_n$ est extrémal (ou sommet de S_n) si l'on a :

$$\forall t \in]0, 1[, \forall (y, z) \in S_n^2 : x = (1-t)y + tz \implies x = y = z$$

b) Fonctions convexes

Définitions :

Soit $f : C \longrightarrow \mathbb{R}$ une fonction et C un sous ensemble convexe de \mathbb{R}^n .

- f est dite convexe sur C si et seulement si l'inégalité suivante est satisfaite :

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

- f est dite strictement convexe sur C si et seulement si :

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y), \quad \forall \lambda \in]0, 1[, \quad \forall x, y \in C \text{ et } x \neq y.$$

- f est dite fortement convexe sur C s'il existe $\alpha > 0$, tel que :

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

(on dit aussi que f est α -convexe).

Caractérisation d'une fonction convexe différentiable

On se place dans l'espace des vecteurs réels \mathbb{R}^n muni de la norme Euclidienne

$$\|x\| = \sqrt{\langle x, x \rangle} \text{ où } \langle x, x \rangle = x^t x = \sum_{i=1}^n x_i^2, \forall x \in \mathbb{R}^n,$$

est le produit scalaire associé et x^t le vecteur transposé de x .

Etant donné $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction réelle deux fois différentiable, le gradient de f noté ∇f est le vecteur colonne défini par les éléments

$$(\nabla f(x))_i = \frac{\partial f(x)}{\partial x_i}, \quad i = 1, \dots, n$$

Le Hessien de f noté $\nabla^2 f$ est la matrice de $\mathbb{R}^{n \times n}$ définie par les éléments

$$(\nabla^2 f(x))_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, n$$

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

- f est convexe si et seulement si :

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

- f est convexe si et seulement si :

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

De plus, f est dite strictement convexe si l'une ou l'autre des inégalités précédentes sont strictes pour $x \neq y$.

- f est fortement convexe si et seulement s'il existe $\alpha > 0$, tel que :

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2, \quad \forall x, y \in C.$$

Si $f \in C^2(C)$, alors :

- f est convexe si et seulement si $\nabla^2 f(x)$ (le Hessien de f) est semi défini positif sur C (c'est à dire que $y^t \nabla^2 f(x) y \geq 0, \forall x, y \in C$ ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont positives).
- f est strictement convexe si et seulement si $\nabla^2 f(x)$ est défini positif sur C (c'est à dire que $y^t \nabla^2 f(x) y > 0, \forall x, y \in C$ et $y \neq 0$ ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives).

1.1 Programmation mathématique

La programmation mathématique constitue un domaine vaste et riche dans l'analyse numérique et traduit plusieurs problèmes pratiques importants.

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

$$(PM) \quad \left\{ \begin{array}{l} \min f(x) \\ x \in C \end{array} \right. \quad \text{où} \quad C = \left\{ \begin{array}{l} x \in \mathbb{R}^n / g_i(x) \leq 0, \quad i = 1, \dots, m \\ h_j(x) = 0, \quad j = 1, \dots, p \end{array} \right\}$$

et f, g_i, h_j sont des fonctions données de \mathbb{R}^n vers \mathbb{R} .

On appelle f la fonction objectif et C l'ensemble des solutions réalisables.

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

parmi les cas particuliers les plus étudiés on note :

- La programmation linéaire (f linéaire, g_i, h_j affines, C h'ortant positif).
- La programmation convexe (f, g_i convexes, h_j affine, C convexe).
- La programmation en nombres entiers (C discret).

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

a) Qualification des contraintes :

Une contrainte g_i est dite active (ou saturée) en $\bar{x} \in C$ si $g_i(\bar{x}) = 0$.

Un point $\bar{x} \in C$ est dit régulier (on dit également que les contraintes sont qualifiées en \bar{x}) si les gradients des contraintes saturées en \bar{x} sont linéairement indépendants.

Il existe aussi deux critères usuels de qualification en tout point de C , à savoir :

- Si toutes les contraintes sont affines.
- Si C est défini uniquement par des inégalités, on a le critère de Slater suivant : $g_i(x)$ est convexe pour tout $i = 1 \dots m$ et qu'il existe un point x^0 tel que $g_i(x^0) < 0$, ($\text{int}(C) \neq \emptyset$).

b) Conditions nécessaires d'optimalité :

Théorème 1.1.1 (*Karuch - Khun - Tucker*) :

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

Si \bar{x} est un optimum local, alors il existe des multiplicateurs $\mu_i \in \mathbb{R}^+, i = 1, \dots, m$ et $\lambda_j \in \mathbb{R}, j = 1, \dots, p$ tel que :

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

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

Les problèmes avec contraintes de type (PM) peuvent souvent être transformés en problèmes sans contraintes à l'aide des multiplicateurs de Karuch Khun Tucker (*KKT*). Cette méthode qui nous permet de trouver les points stationnaires \bar{x} de la fonction f ,

revient en effet à associer à chaque contrainte un scalaire inconnu μ_i, λ_j . On forme par la suite une combinaison linéaire des gradients de la fonction objectif et des contraintes comme le montre le système ci-dessus.

Dans le cas particulier où $m = 0$, les conditions précédentes sont appelées les conditions de Lagrange et s'écrivent comme suit :

Si \bar{x} est un optimum local, alors il existe $\lambda_j \in \mathbb{R}, j = 1 \dots p$ tel que :

$$\begin{cases} \nabla f(\bar{x}) + \sum_{j=1}^p \lambda_j \nabla h_j(\bar{x}) = 0 \\ h_j(\bar{x}) = 0 \quad j = 1, \dots, p. \end{cases}$$

Application (Projection sur un hyperplan) :

Soit $H = \{x \in \mathbb{R}^n / a^T x = \alpha\}$ (où $0 \neq a \in \mathbb{R}^n$ et $\alpha \in \mathbb{R}$) un hyperplan de \mathbb{R}^n et $y \in \mathbb{R}^n \setminus H$.

H étant convexe fermé et non vide, l'existence et l'unicité du projeté orthogonal de y sur H sont assurées par le Théorème de séparation.

En termes de programmation mathématique, le problème convexe différentiable suivant admet une solution unique

$$(P_H) \quad \begin{cases} \min (\frac{1}{2} \|x - y\|^2) \\ x \in H. \end{cases}$$

Laquelle est parfaitement caractérisée par les conditions de (KKT).

$$\bar{y} = Proj_H(y) \iff \bar{y} = y - \left(\frac{a^T y - \alpha}{\|a\|^2} \right) a$$

1.2 Programmation linéaire

La programmation linéaire dans \mathbb{R}^n traite la résolution des problèmes d'optimisation pour lesquels la fonction objectif et les contraintes sont linéaires, les spécialistes la

considèrent comme étant la technique la plus utilisée dans la recherche opérationnelle.

Un programme linéaire peut être écrit sous différentes formes équivalentes, la plus utilisée en pratique est la forme standard :

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

avec $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et A une matrice de $\mathbb{R}^{m \times n}$ donnés.

Le programme linéaire (PL) caractérisé par :

$$(PL) \begin{cases} \min c^t x \\ Ax \leq b, \quad (\geq b) \\ x \geq 0 \end{cases}$$

représente sa forme canonique.

Dans toute la suite, on s'intéresse au programme linéaire sous sa forme standard.

On peut toujours passer d'une forme à une autre en introduisant des variables d'écart comme suit :

a- Ramener (PL) de la forme standard à la forme canonique par l'équivalence suivant :

$$Ax = b \iff Ax \leq b \text{ et } Ax \geq b$$

b- Ramener (PL) de la forme canonique à la forme standard comme suit :

$$Ax \leq b \implies Ax + y = b \text{ avec } y \geq 0 \text{ ou } : Ax \geq b \implies Ax - y = b \text{ avec } y \geq 0$$

- Un vecteur x vérifiant $P = \{Ax = b \text{ et } x \geq 0\}$ est appelé solution réalisable de (PL) .

- Un vecteur x vérifiant $int(P) = \{Ax = b \text{ et } x > 0\}$ est appelé solution strictement réalisable de (PL) .

- Un vecteur $x \in P$ vérifiant (PL) est appelé solution optimale de (PL) .

- Un programme linéaire (PL) réalisable est borné si l'objectif est borné sur P .
- Un point x est un point extrême de P si et seulement si les colonnes $\{A^j : x_j > 0\}$ de A sont linéairement indépendantes, on en déduit que si P est non vide. Alors il existe au moins un point extrême.

Chapitre 2

Méthode de Trajectoire Centrale pour la programmation linéaire

2.1 Présentation de la méthode classique

A tout problème (PL) on associe le problème pénalisé suivant :

$$(PL)_\mu \left\{ \begin{array}{l} \min f_\mu(x) \\ s.c \\ Ax = b \\ x > 0, \end{array} \right.$$

où f_μ est la fonction pénalisée définie par :

$$f_\mu(x) = c^t x - \mu \sum_{i=1}^n \log(x_i).$$

et μ est un paramètre barrière strictement positif.

Lemme 2.1.1 *La fonction f_μ est strictement convexe.*

Preuve. La fonction f_μ est écrite sous la forme :

$$f_\mu(x) = c^t x - \mu \sum_{i=1}^n \log(x_i)$$

En effet, $f_\mu \in C^\infty$ et nous avons en particulier

$$\nabla f_\mu(x) = c - \mu X^{-1} e,$$

avec $e = (1, \dots, 1)^t \in \mathbb{R}^n$.

$$\nabla^2 f_\mu(x) = \mu X^{-2},$$

où $X = \text{diag}(x_1, \dots, x_n)$ est une matrice diagonale définie positive, car $x_i > 0$, et $\mu > 0$, alors $\nabla^2 f_\mu(x)$ est une matrice définie positive, d'où le résultat. ■

1. Si $\overset{0}{\mathcal{F}}_{(PL)}$ et $\overset{0}{\mathcal{F}}_{(DL)}$ sont non vides, alors pour tout $\mu > 0$, le problème $(PL)_\mu$ admet une solution unique, notée $x(\mu)$, et appelée "point central".
2. Quand $\mu \rightarrow 0$, $x(\mu) \rightarrow x^*$ solution optimale de (PL) .
3. La fonction $\mu \rightarrow (x(\mu), y(\mu), s(\mu))$ définit la trajectoire centrale qu'on note

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

T_C s'appelle trajectoire centrale de $(PL)_\mu$.

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

$$\begin{cases} c - \mu X^{-1} e - A^t y & = 0 \\ Ax & = b, \end{cases}$$

où $y \in \mathbb{R}^m$ est le multiplicateur de Lagrange associé à la contrainte $Ax = b$ du problème $(PL)_\mu$.

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

$$(S_\mu) \begin{cases} Xs & = \mu e \\ Ax & = b, x > 0 \\ A^t y + s & = c, s > 0. \end{cases}$$

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

Le système (S_μ) désigne aussi les conditions d'optimalité du problème dual paramétré $(DL)_\mu$ suivant :

$$(DL)_\mu \begin{cases} \max b^t y + \mu \sum_{i=1}^n \log(s_i) \\ s.c \\ A^t y + s = c, \\ s > 0. \end{cases}$$

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

$$(S'_\mu) \begin{cases} b - Ax & = 0 \\ \mu S^{-1}e - x & = 0 \\ A^t y + s & = c, \end{cases}$$

où x est le multiplicateur de Lagrange associé à la contrainte $A^t y + s = c$ et $S = \text{diag}(s_1, \dots, s_n)$. D'où (S'_μ) est équivalent à (S_μ) .

Le système (S_μ) est un système d'équations non linéaires, pour cela, la méthode de Newton est l'une des méthodes les plus utilisées pour sa résolution. A chaque μ , et en appliquant la méthode de newton, on trouve une solution $(x(\mu), y(\mu), s(\mu))$ proche de la trajectoire centrale T_C (condition de proximité).

2.1.1 Concept de proximité

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

$$T_C(\theta) = \left\{ (x, y, s) \in \overset{0}{\mathcal{F}}_{(PL)} \times \overset{0}{\mathcal{F}}_{(DL)} / \|X(\mu)S(\mu) - \mu e\|_2 \leq \theta\mu, 0 < \theta < 1 \right\}$$

En calculant une racine de la fonction $F(x, y, s) = 0$, issue du système (S_μ) , et définie par :

$$F(x, y, s) = \begin{pmatrix} XSe - \mu e \\ Ax - b \\ A^t y + s - c \end{pmatrix}$$

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

$$(x^+, y^+, s^+) = (x, y, s) + \alpha(dx, dy, ds), \quad \alpha \in]0, 1].$$

où (dx, dy, ds) est solution du système linéaire :

$$\nabla F(x, y, s) \begin{pmatrix} dx \\ dy \\ ds \end{pmatrix} = -F(x, y, s) \quad (SN)$$

(SN) s'écrit sous forme matricielle comme suit :

$$\begin{bmatrix} S & 0 & X \\ A & 0 & 0 \\ 0 & A^t & I \end{bmatrix} \begin{bmatrix} dx \\ dy \\ ds \end{bmatrix} = - \begin{bmatrix} XSe - \mu e \\ 0 \\ 0 \end{bmatrix} \quad (SN_\mu)$$

Théoriquement, on suppose que la solution initiale est strictement réalisable, qu'elle soit proche de la trajectoire centrale, mais l'obtention de ce point est une difficulté majeure. Pour résoudre ce problème, Kebbiche et al. [7] ont introduit le paramètre poids,

associé à la fonction barrière.

2.2 Méthode de Trajectoire Centrale avec poids pour la programmation linéaire

2.2.1 Présentation de la méthode

On considère le problème linéaire sous forme standard suivant :

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

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

$$(PL)_{\mu r} \begin{cases} \min f_{\mu r}(x) = c^t x - \mu \sum_{i=1}^n r_i \log x_i \\ Ax = b \\ x > 0, \mu > 0, \end{cases}$$

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

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

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

Où encore sous la forme :

$$(S_{\mu r}) \begin{cases} A^t y + s & = c \\ Ax & = b \\ Xs - \mu r & = 0 \\ (x, s, \mu) & > 0, \end{cases}$$

où l'on a posé : $s = \mu X^{-1}r$.

Pour $r = e$ dans le système $(S_{\mu r})$, on trouve le système classique, c'est à dire la méthode primale-duale de trajectoire centrale classique (sans poids).

On applique alors la méthode de Newton pour $(x, y, s) \in \overset{0}{\mathcal{F}}_{(PL)} \times \overset{0}{\mathcal{F}}_{(DL)}$, on obtient le système :

$$(SN_{\mu r}) \begin{cases} Adx & = 0 \\ A^t dy + ds & = 0 \\ Xds + Sdx & = -Xs + \mu r, \end{cases}$$

dont la solution est (dx, dy, ds) . Le nouvel itéré est :

$$(x^+, y^+, s^+) = (x, y, s) + \alpha(dx, dy, ds), \quad \alpha \in]0, 1].$$

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

$$T_r(\theta) = \{(x, y, s) \in \overset{0}{\mathcal{F}}_{(PL)} \times \overset{0}{\mathcal{F}}_{(DL)} / \|XSe - \mu r\| \leq \theta\mu, 0 < \theta < 1\}.$$

Algorithme 2. Méthode de trajectoire centrale pour la programmation linéaire

Début algorithme

Initialisation

$$k = 0, (x^0, y^0, s^0) \in \mathcal{F}_{(PL)}^0 \times \mathcal{F}_{(DL)}^0;$$

Un paramètre de précision $\epsilon > 0$;

$$\varrho = \frac{\|X^0 S^0 e\|}{\sqrt{n}}, r = \frac{X^0 S^0 e}{\varrho}, \eta = \min\{r_i\}, \theta = \left(1 - \frac{\sqrt{2}}{2}\right)\eta, \beta = 1 - \frac{\theta}{\sqrt{n}} \text{ et } \mu^0 = \frac{(x^0)^t s^0}{n};$$

Si $\|X^0(\mu^0)s^0(\mu^0) - \mu^0 e\|_2 \leq \mu^0 \theta$ alors :

Aller à $T_c(SP)$ (trajectoire centrale sans poids)

Sinon :

Aller à $T_c(AP)$ (trajectoire centrale avec poids)

Fin si

$T_c(SP)$:

$$k = 0$$

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

Etape 1 : $\mu^{k+1} = \beta \mu^k = \beta \frac{(x^k)^t s^k}{n}$.

Etape 2 : Calculer la direction (d_x^k, d_y^k, d_s^k) , en résolvant le système (SN_μ)

Etape 3 : Calculer le pas de déplacement α_k

Etape 4 : Calculer : $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (d_x^k, d_y^k, d_s^k)$

$$k = k + 1$$

Fin tant que ;

$T_c(AP)$:

$$k = 0$$

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

Etape 1 : $\mu^{k+1} = \beta \mu^k = \beta \frac{(x^k)^t R s^k}{n}$.

Etape 2 : Calculer la direction (d_x^k, d_y^k, d_s^k) , en résolvant le système $(SN_{\mu r})$

Etape 3 : Calculer le pas de déplacement α_k

Etape 4 : Calculer : $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (d_x^k, d_y^k, d_s^k)$

$$k = k + 1$$

Fin tant que ;

Fin algorithme.

2.2.2 Etude de la convergence

L'étude de la convergence a été établie par Z. Kebbiche et al. [7] qui ont montré sous certaines conditions que les itérés restent au voisinage de la trajectoire centrale et ils ont obtenu, plus précisément, les résultats suivants :

Théorème 2.2.1 [7] Soit $\theta = \delta = \left(1 - \frac{\sqrt{2}}{2}\right) \eta$,

Pour $(x, y, s) \in T_r(\theta)$ et $\mu^+ = \beta\mu$,

tel que $\beta = \left(1 - \frac{\delta}{\sqrt{n}}\right)$, $\mu = \frac{x^t R s}{n}$ alors on a :

1) $(x^+, y^+, s^+) \in T_r(\theta)$.

2) $\frac{x^{+t} R s^+}{n} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) \frac{x^t R s}{n}$.

Chapitre 3

Présentation et principe de la méthode pour minimiser une fonction convexe non linéaire

L'objectif de ce chapitre est de présenter une méthode primale-duale de type trajectoire centrale pour minimiser une fonction convexe non linéaire sous contraintes linéaires, il s'agit d'une approche barrière logarithmique avec poids qui offre une pénibilité dans le choix du point initial.

On traite alors les conditions de KKT qui sont nécessaires et suffisantes par la méthode

De Newton pour obtenir le nouvel itéré, l'algorithme proposé peut démarrer par un point Strictement réalisable et qui appartient au voisinage de la trajectoire centrale.

3.1 Méthode de la trajectoire centrale classique

3.1.1 Description de la méthode

On considère le problème d'optimisation non linéaire convexe suivant :

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

Où $f : \mathbb{R}^n \rightarrow \mathbb{R}$; est une fonction non linéaire, convexe et différentiable, A est une matrice de type (m, n) et de plein rang ($rgA = m < n$) et $b \in \mathbb{R}^m$.

On note par :

$\mathcal{F} = \{x \in \mathbb{R}_+^n : Ax = b\}$,L'ensemble des solutions réalisables de (PC) .

$\mathring{\mathcal{F}} = \{x \in \mathbb{R}_{++}^n : Ax = b\}$,L'ensemble des solutions strictement réalisables de (PC) .

On associe à (PC) le problème perturbé suivant :

$$(PC)_\mu \begin{cases} \min f_\mu(x) = \left(f(x) - \mu \sum_{i=1}^n \ln x_i \right) \\ Ax = b \\ x > 0 \end{cases}$$

$\mu > 0$ est le paramètre barrière.

f_μ est une fonction strictement convexe, il suffit de voir que :

$$\nabla f_\mu(x) = \nabla f(x) - \mu X^{-1}e, \text{ avec } X = \text{diag}(x_1, \dots, x_n)$$

$$\nabla^2 f_\mu(x) = \nabla^2 f(x) + \mu X^{-2} \text{ est une matrice définie positive (puisque } f \text{ est convexe).}$$

$(PC)_\mu$ est un problème strictement convexe (les contraintes sont linéaires), donc les conditions d'optimalité de KKT sont nécessaires et suffisantes, elles s'écrivent comme suit : $(PC)_\mu$ est un problème strictement convexe (les contraintes sont linéaires), donc les conditions d'optimalité de KKT sont nécessaires et suffisantes, elles s'écrivent comme Suit :

$$\begin{cases} A^t y + s = \nabla f_\mu(x) \\ Ax = b \\ Xs - \mu e = 0 \\ (x, s) > 0 \end{cases} \quad (3.1)$$

Pour chaque valeur de μ , on résout le système (3.1) par la méthode de Newton, on obtient ainsi une suite de solutions paramétrisées qui converge vers la solution du problème initial lorsque μ tend vers 0.

On définit l'ensemble :

$$T = \{(x, y, s) \in \mathcal{F} \times \mathbb{R}_{++}^n : A^t y + s = \nabla f_\mu(x)\} \quad (3.2)$$

La trajectoire centrale est l'ensemble de toutes les solutions $\{(x_\mu, y_\mu, s_\mu) / \mu > 0\}$ du système :

$$\begin{cases} A^t y + s = \nabla f_\mu(x) \\ Ax = b \\ x_i s_i = \mu, i = 1, \dots, n \\ (x, s) > 0 \end{cases} \quad (3.3)$$

Elle est notée par :

$$TC = \{(x, y, s) \in T : XSe - \mu e = 0\}$$

Puisqu'on a une équation non linéaire dans le système (3.3), il est difficile d'obtenir une solution exacte de $(PC)_\mu$. Nous allons donc nous contenter d'une solution approximative dont la qualité est jugée satisfaisante lorsqu'elle se trouve dans le voisinage de la trajectoire centrale (TC) . Un point est dit voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$T(\theta) = \{(x, y, s) \in T : \| XSe - \mu e \| \leq \theta\mu, 0 < \theta < 1\}$$

Les méthodes primales-duales déterminent les solutions du système (3.3) en appliquant la méthode de Newton comme suit : on définit la fonction non linéaire :

$$F_\mu : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$$

$$F_\mu(x, y, s) = \begin{pmatrix} A^t y + s - \nabla f(x) \\ Ax - b \\ XSe - \mu e \end{pmatrix} \quad (3.4)$$

Où $X = \text{diag}(x_1, \dots, x_n)$, $S = \text{diag}(s_1, \dots, s_n)$.

La direction de Newton $d\omega = (dx, dy, ds)$ est solution du système linéaire :

$$\nabla F_\mu d\omega = -F_\mu \quad (3.5)$$

Où

$$\nabla F_\mu = \begin{pmatrix} A & 0 & 0 \\ \nabla^2 f(x) & -A^t & -I \\ S & 0 & X \end{pmatrix} \quad (3.6)$$

Le nouvel itéré est :

$$(x^+, y^+, s^+) = (x, y, s) + \alpha(dx, dy, ds)$$

Si le point initial calculé n'est pas voisin de la trajectoire centrale, rien ne garantit la convergence de cet algorithme même si ce point est strictement réalisable.

3.2 Méthode de trajectoire centrale avec poids

3.2.1 Présentation de la méthode :

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

$$(PC)_{\mu\tau} \begin{cases} \min f_{\mu\tau}(x) = \left(f(x) - \mu \sum_{i=1}^n \tau_i \ln x_i \right) \\ Ax = b \\ x > 0 \end{cases} \quad (3.7)$$

$\mu > 0$, $\tau = (\tau_1, \tau_2, \dots, \tau_n)^t \in \mathbb{R}_+^n$ est le vecteur poids associé à la fonction barrière.

$f_{\mu\tau}(x)$ est strictement convexe et les contraintes sont linéaires, alors les conditions de KKT correspondantes sont nécessaires et suffisantes et s'écrivent comme suit :

$$\begin{cases} \nabla f(x) - \mu X^{-1}\tau - A^t y = 0 \\ Ax = b \\ x > 0, y \in \mathbb{R}^m \end{cases} \quad (3.8)$$

$$\Leftrightarrow \begin{cases} A^t y + s = \nabla f(x) \\ Ax = b \\ Xs - \mu\tau = 0 \\ (x, s) > 0 \end{cases} \quad (3.9)$$

Où $s = \mu X^{-1}\tau$.

Pour $\tau = e$ dans le système (3.9), on trouve le système classique (3.3), c'est à dire la méthode primale-duale de trajectoire centrale classique (sans poids).

Pour résoudre le système (3.9), on applique la méthode de Newton. A chaque itération, on résout le système linéaire (3.10) suivant :

$$\begin{pmatrix} A & 0 & 0 \\ \nabla^2 f(x) & -A^t & -1 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ ds \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XSe + \mu\tau \end{pmatrix} \quad (3.10)$$

Le nouvel itéré :

$$(x^+, y^+, s^+) = (x, y, s) + (dx, dy, ds).$$

Le point (x^+, y^+, s^+) qui vérifie le système (2.9) est voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$T_\tau(\theta) = \{(x, y, s) \in T : \|XSe - \mu\tau\| \leq \theta\mu, 0 < \theta < 1\} \quad (3.11)$$

3.2.2 Calcul de la direction de descente (dx, dy, ds)

Le calcul de la direction nécessite la résolution du système linéaire (2.10) qui s'écrit sous la forme :

$$\begin{cases} -\nabla^2 f(x) dx + A^t dy + Ids = 0 \\ Adx = 0 \\ Sdx + Xds = X^k S^k e - \mu^k \tau \end{cases} \quad (3.12)$$

On peut éliminer ds grâce à la dernière équation, qui donne :

$$ds = -X^{-1}Sdx + \mu X^{-1}\tau - s$$

le système (3.12) se réduit au système :

$$\begin{pmatrix} X^{-1}S + \nabla^2 f(x) & -A^t \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \mu X^{-1}\tau - s \\ 0 \end{pmatrix} \quad (3.13)$$

On peut poursuivre la réduction de la dimension du système linéaire (3.13) à résoudre

en éliminant dx grace à la première équation

$$dx = (\nabla^2 f(x) + X^{-1}S)^{-1} A^t dy + (\nabla^2 f(x) + X^{-1}S)^{-1} (\mu X^{-1}\tau - s)$$

On obtient alors le système linéaire suivant :

$$A (\nabla^2 f(x) + X^{-1}S)^{-1} A^t dy = -A (\nabla^2 f(x) + X^{-1}S)^{-1} (\mu X^{-1}\tau - s) \quad (3.14)$$

On résout le système (3.14), pour obtenir la direction dy , puis on déduit dx et ds .

Remarque 3.2.1 *Comme les contraintes sont linéaires, le point initial (x^0, y^0, s^0) se calcule de la même manière que dans la programmation linéaire.*

Remarque 3.2.2 *Nous avons introduit un pas variable dans l'itération*

$$(x^+, y^+, s^+) = (x, y, s) + \alpha_k (dx, dy, ds)$$

3.2.3 Conclusion

On a présenté dans ce travail une étude théorique, une analyse algorithmique des méthodes de trajectoire centrale pour minimiser une fonction convexe sur les contraintes linéaires avec la méthode de trajectoire centrale classique et avec poids. Dans des études futures, on va entamer une étude algorithmique et numérique pour donner des solutions approximatives. Aussi, on parlera des types de convergences de cette méthode ainsi que ses avantages et ses inconvénients.

Bibliographie

- [1] E. D. Andersen, J. Gondzio, Cs. Mészáros, X. Xu, *Implementation of interior point methods for large scale linear programming*, in : T. Terlaky (Ed.), *Interior Point Methods of Mathematical Programming*, Kluwer Academic Publisher, The Netherlands, 189–252, (1996).
- [2] I. Adler, M.G.C. Resende, G.Veiga, *An implementation of Karmarkar's algorithm for linear programming*. Operations Research Center, Report 86-8, University of California at Berkeley, (1989).
- [3] M. Bazarra, H.D. Sherali, C.M. Shetty, *Non linear programming theory and algorithms*, Second edition (1993).
- [4] G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, (1985).
- [5] J.C. Culioli, *Introduction à l'optimisation*, Ellipses (1994).
- [6] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, *Combinatorica*, 4, 373–395, (1984).
- [7] Z.Kebbiche, A. Keraghel & A. Yassine, *Extension of a projective interior point method for linearly constrained convex programming*, *Applied Mathematics and Computation* Vol 193, Issue2, 553–559, 1 November (2007).
- [8] A. Keraghel, *Analyse convexe : Théorie fondamentale et exercices*, Editions Dar el'Houda, Ain M'lila, Algérie, (1999).

- [9] A. Keraghel, *Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar*, Dissertation thesis, Université Joseph Fourier, Grenoble, France (1989).
- [10] A. Keraghel, D. Benterki, *Sur les performances de l'algorithme de Karmarkar pour la programmation linéaire*, Revue Roumaine des sciences techniques et Mécanique appliquées, Tome 46, n°1, 87–96, (2001).
- [11] A. Leulmi, *Une procédure améliorante d'une méthode projective en programmation linéaire*. Thèse de magister, Université de Skikda. Algérie. (2007).
- [12] I.J. Lustig, *A practical approach to Karmarkar's algorithm*, Technical report sol 85-5 Systems optimization laboratory, dep of operations res. STANFORD. univ ; STANFORD CALIFORNIA 94305 (1985).
- [13] R. Naseri , A. Valinejad, *An extended variant of Karmarkar's interior point algorithm* , Applied Mathematics and computation 184 737–742 (2007).
- [14] R.T. Rockafellar, *Convex analysis*, published by Princeton University Press, 41 William Street, Princeton, New Jersey 08540.
- [15] C. Roos,T. Terlaki, J. Vial, *Optimization Theory and Algorithm for Linear Programming Optimization*,Princeton University, (2001)