

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohammed Seddik Ben Yahia-Jijel



Faculté des Sciences Exactes et Informatique
Département de mathématiques

Mémoire de fin d'études

Présenté pour l'obtention du diplôme de

Master

Spécialité : Mathématiques

Option : EDP et Applications

Thème

**La Méthode de PI primale-duale basée sur une
fonction barrière à deux paramètres pour PL**

Présenté par :

Benadouane Imane

Devant le jury :

Président	T. Haddad	Prof Université de Jijel
Encadreur	I.Touil	MCB Université de Jijel
Examineur	L. Menniche	MCB Université de Jijel

Promotion : 2019-2020

Remerciements

Un remerciement chaleureux et spécial à Mme. Touil Imene pour avoir accepté de m'encadrer, m'avoir proposé ce sujet, m'avoir dirigé et accompagné durant toute cette période de travail.

Nous remercions Messieurs et Mmes nom et nom d'avoir accepté de faire partie du jury.

Je souhaite adresser mes remerciements les plus sincères à tous mes enseignants au niveau du département de mathématiques et en particulier à ceux du master EDP.

Mes remerciements vont vers tous mes collègues, amis et toutes les personnes qui m'ont soutenue et encouragée.

Finalement, je ne peux pas oublier ma chère famille. Ce travail est aussi un hommage destiné à les remercier pour tous ses efforts.

Dédicace

À

Mes chères parents,

Mes frères ,

Ma grande famille,

Mes amies ,

Tous mes collègues de la promotion 2019-2020,

Tous ceux qui sont proches à mon coeur..

B.Imane.

Table des matières

Introduction	iii
1 <i>Préliminaires et notions fondamentales</i>	4
1.1 Analyse convexe	4
1.1.1 Notions de convexité	4
1.1.2 Caractérisation d'une fonction convexe différentiable	7
1.1.3 Fonction barrière	8
1.1.4 Formules de Taylor	9
1.2 Programmation mathématique	10
1.2.1 Définitions	10
1.2.2 Classification des problèmes	11
1.2.3 Principaux résultats d'existence	12
1.2.4 Conditions d'optimalité	12
1.3 Programmation linéaire (PL)	13
1.3.1 Les formes usuelles d'un programme linéaire	14
1.3.2 Relation entre forme standard et forme canonique	17
1.3.3 Dualité Lagrangienne	17
1.3.4 Relation primal-dual pour la programmation linéaire	18
1.4 Algorithme d'optimisation	20
1.4.1 Description	21
1.4.2 Convergence	21

1.4.3	Taux de convergence	22
2	<i>Résolution du problème d'optimisation linéaire via les fonctions noyau</i>	23
2.1	Propriétés de la nouvelle fonction noyau	28
2.1.1	Éligibilité de la nouvelle fonction noyau	30
2.1.2	Détermination du pas de déplacement	36
2.2	Nombre d'itérations de l'algorithme	41
2.2.1	Nombre d'itérations internes	41
2.2.2	Nombre total d'itérations	42
3	<i>Tests numériques</i>	45
3.1	Choix des paramètres p et q	46
3.1.1	commentaires	51
3.2	Comparaison des algorithmes	52
3.2.1	commentaires	54
	Bibliography	55

Notations générales

KKT	: Karush-Kuhn-Tucker .
PM	: Programme Mathématique .
PL	: Programme Linéaire .
e	: Vecteur tel que $e = (1, \dots, 1)^t \in \mathbb{R}^n$.
P	: Programme linéaire primal .
D	: Programme linéaire dual .
P_μ	: Problème perturbé Primal .
D_μ	: Problème perturbé Dual .
$MPIs$: Méthodes de Points Intérieurs .
CPI	: Condition de Points Intérieurs .
NT	: Direction de Nesterov et Todd .
$CNS(P_\mu)$: Conditions Nécessaires et Suffisantes (ou de KKT) sur (P_μ) .
$CNS(D_\mu)$: Conditions Nécessaires et Suffisantes (ou de KKT) sur (D_μ) .
$\ \cdot\ $: Norme Euclidienne définie par $\ x\ = (\sum_i x_i^2)^{\frac{1}{2}}$.
c-à-d	: c'est à dire.
i.e.,	: identiquement.

Introduction

L'optimisation est un outil important pour les sciences de la décision et l'analyse des systèmes physiques. Les ingénieurs, les économistes, les décideurs se heurtent quotidiennement, quel que soit leur secteur d'activité, à des problèmes d'optimisation. Il peut s'agir de minimiser un coût de production, d'optimiser le parcours d'un véhicule ou le rendement d'un portefeuille boursier, de rationaliser l'utilisation de ressources, d'améliorer les performances d'un circuit électronique, de fournir une aide à la décision à des managers, etc. Un problème d'optimisation consiste à déterminer la plus petite (ou grande) valeur possible qu'une fonction réelle $f : C \rightarrow \mathbb{R}$ nommée *fonction objectif* puisse prendre dans l'ensemble C nommé *ensemble des solutions réalisables* ou aussi *ensemble des contraintes*. Les problèmes d'optimisation sont divisés naturellement en deux filières :

- Les modèles d'optimisation stochastiques.
- Les modèles d'optimisation déterministe.

L'optimisation déterministe à son tour se divise à deux branches, ceux avec des variables réelles, qui sont appelées souvent problèmes d'optimisation continue, et ceux avec des variables discrètes, qui sont appelées souvent problèmes d'optimisation combinatoire.

Dans l'optimisation continue on considère l'ensemble des solutions réalisables comme un ensemble réel, tandis qu'en optimisation combinatoire on cherche un certain objectif dans des ensembles finis ou infinis dénombrables.

Ce mémoire se cadre dans le contexte d'optimisation (ou problème de programmation) linéaire, branche de l'optimisation continue, elle s'applique surtout en gestion (gestion de

production) et en économie appliquée. Parmi de tels problèmes, on peut citer les problèmes de transport, gestion des banques, les industries lourdes et légères, l'agriculture, les chaînes commerciales, et même le domaine des applications militaires, etc.

Les méthodes de résolution d'un problème de programmation linéaire sont : les méthodes simpliciales et les méthodes de points intérieurs.

On désigne par méthode de point intérieur, toute procédure de résolution générant une suite de points appartenant à l'intérieur (relatif) du domaine réalisable (admissible) et convergent vers une solution optimale du programme considéré. Il y a principalement trois grandes catégories des méthodes de points intérieurs : les méthodes affines, les méthodes de réduction du potentiel et les méthodes de trajectoire centrale.

Dans notre étude, nous avons pris comme modèle la méthode de trajectoire centrale primale-duale, à cause de ses avantages par rapport aux autres classes de méthodes de points intérieurs. Nous avons proposé un algorithme contenant des modifications consistantes par l'introduction des nouvelles techniques parmi lesquelles les fonctions noyau pour trouver une classe de directions dans le but d'améliorer le comportement de l'algorithme (la complexité algorithmique).

Dans la programmation linéaire, on s'intéresse à des notions comme la convexité, la différentiabilité, l'existence et l'unicité d'une solution, les conditions nécessaires, les conditions suffisantes, et les conditions nécessaires et suffisantes d'optimisation. Ces notions sont présentées dans le premier chapitre de ce mémoire.

Le deuxième chapitre est consacré à la méthode de trajectoire centrale pour la programmation linéaire via la fonction noyau proposée par Bouafia et al. [5], dans le but de trouver une nouvelle classe de directions, puis en calculant le pas de déplacement par l'utilisation d'une procédure dite de positivité. Signalant que le calcul de la direction de

discente et le pas de déplacement jouent un rôle important dans le comportement de l'algorithme (vitesse de convergence, nombres des itérations). On termine ce chapitre par l'analyse de la complexité de notre algorithme.

Enfin, l'algorithme associé à cette méthode a été programmé sur Matlab, les résultats obtenus ont été présentés au troisième chapitre de ce mémoire d'ordre comparatif entre quelques fonctions noyau.

Chapitre 1

Préliminaires et notions fondamentales

1.1 Analyse convexe

1.1.1 Notions de convexité

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

Définition 1.1.1 *On dit que l'ensemble $C \subset \mathbb{R}^n$ est convexe si :*

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

De plus :

- *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 :

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

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

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

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

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

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

Définition 1.1.2 • Une fonction f définie sur un ensemble convexe C est dite convexe si :

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

- f est dite strictement convexe sur un ensemble convexe C si :

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

- On dit que f est (Strictement) convexe sur un ensemble convexe C si $(-f)$ est (Strictement) concave sur C .
- f est dite mid-convexe sur C si :

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

- f est dite quasi-convexe sur C si :

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

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

$$\forall \lambda \in]0, 1[, \forall x, y \in C \quad \text{et} \quad x \neq y,$$

on a

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2}\alpha\lambda(1 - \lambda)\|x - y\|^2.$$

(On dit aussi que f est α -convexe).

- f est convexe sur C si et seulement si f est mid- et quasi-convexe sur C .
- Si f est une fonction continue sur un convexe C , On a :

— f est convexe sur C si seulement si f est mid-convexe C .

— f est α -convexe sur C si seulement si :

$$\forall x, y \in C, f\left(\frac{x+y}{2}\right) \leq \frac{f(x)+f(y)}{2} - \frac{\alpha}{8}\|x-y\|^2.$$

Définition 1.1.3 Une fonction f est dite coercive sur un ensemble convexe C si :

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty, \forall x \in C.$$

Définition 1.1.4 On dit que l'ensemble $A \subset \mathbb{R}^n$ est affine si :

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

Définition 1.1.5 Une fonction f définie de \mathbb{R}^n dans \mathbb{R}^m est dite affine si :

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

Définition 1.1.6 Étant donné $S \subseteq \mathbb{R}^n$, alors il existe une partie affine unique $F \subseteq \mathbb{R}^n$ contenant S appelée enveloppe affine de S et notée $\text{aff}(S)$. Autrement dit $\text{aff}(S)$ est la plus petite partie affine de \mathbb{R}^n contenant S .

$$\text{aff}(S) = \cap \{F_s / F_s \text{ affine et } S \subset F_s\}.$$

- Nous avons par définition $\dim(S) = \dim(\text{aff}(S))$.
- Si $S \neq \emptyset$, alors $\text{aff}(s) \neq \emptyset$ (puisque $S \subseteq \text{aff}(S)$).
- Un sous ensemble S de \mathbb{R}^n est affine si et seulement si $S = \text{aff}(S)$.

Définition 1.1.7 Soit $f : U \subset E \rightarrow F \subset \mathbb{R}$, avec U un ouvert de E , $E \subset \mathbb{R}^n$ et a un point de U , on dit que f est une fonction différentiable en a s'il existe une application linéaire $L \in \mathcal{L}(E, F)$ telle que :

$$\lim_{x \rightarrow a} \frac{\|f(x) - f(a) - L(x-a)\|_F}{\|x-a\|_E} = 0.$$

L'application L est alors unique et est appelée différentielle de f en a , on la note $df(a)$.

Définition 1.1.8 Soit $f : U \subset E \rightarrow F \subset \mathbb{R}$, avec U ouvert, E sous ensemble de \mathbb{R}^n et $a \in U$.

- On dit que f est deux fois différentiable en a si f est différentiable sur un voisinage de a et si df est elle même différentiable en a .
- On dit que f est deux fois différentiable sur U si f est deux fois différentiable en tout point de U .

Définition 1.1.9 Soit f une fonction différentiable en tout point de $U \subset E \subset \mathbb{R}^n$, on peut considérer l'application :

$$\begin{aligned} df &\mapsto \mathcal{L}(E, F) \\ x &\mapsto df(x). \end{aligned}$$

Si df est continue sur U , on dit que f est continûment différentiable sur U , ou encore que f est de classe C^1 sur U .

Définition 1.1.10 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continûment différentiable, leur gradient au point $x \in \mathbb{R}^n$ s'écrit :

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^t.$$

Définition 1.1.11 Soit U un ouvert de \mathbb{R}^n et $f : U \rightarrow \mathbb{R}$ une fonction deux fois différentiable en $a \in U$ dont toutes les dérivées partielles d'ordre deux sont définies en a , alors la matrice :

$$\nabla^2 f(a) = \begin{bmatrix} \frac{\partial^2 f(a)}{\partial x_1^2} & \frac{\partial^2 f(a)}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f(a)}{\partial x_n \partial x_1} \\ \frac{\partial^2 f(a)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(a)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(a)}{\partial x_n \partial x_2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f(a)}{\partial x_1 \partial x_n} & \frac{\partial^2 f(a)}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f(a)}{\partial x_n^2} \end{bmatrix}$$

est appelée matrice Hessienne de f en a .

1.1.2 Caractérisation d'une fonction convexe différentiable

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

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

- Autrement dit : f est convexe si et seulement si :

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

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 $\alpha > 0$, telque :

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

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 $y^t \nabla^2 f(x) y \geq 0, \forall y \in C$ 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 $y^t \nabla^2 f(x) y > 0, \forall y \in C$ et $y \neq 0$ ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives).

1.1.3 Fonction barrière

Définition 1.1.12 *L'intérieur relatif d'un sous ensemble non vide D de \mathbb{R}^n , noté $Int(D)$ est défini comme l'intérieur de D vu comme sous ensemble de $aff(D)$.*

$$Int(D) = \{x \in aff(D) / \exists \varepsilon > 0 : (x + \varepsilon B) \cap aff(D) \subseteq D\},$$

où B est la boule unitaire, i.e, $B = B(0, 1)$.

Si $Int(D) = D$, alors D est dit relativement ouvert.

Définition 1.1.13 *On appelle fonction barrière toute fonction f qui vérifie :*

1. $f(x)$ finie, si x appartient à l'intérieur relatif du domaine réalisable (admissible).
2. $f(x)$ tend vers l'infini quand x s'approche de la frontière du domaine réalisable.

Remarque 1.1.1 *La fonction barrière classique la plus utilisée en programmation mathématique est la fonction barrière logarithmique.*

1.1.4 Formules de Taylor

Soit $\Omega \subset \mathbb{R}^n$ ouvert, $f : \Omega \rightarrow \mathbb{R}^n$, $a \in \Omega$ et $h \in \mathbb{R}^n$ telque $[a, a + h] \subset \Omega$. Alors :

1. Si $f \in C^1(\Omega)$ Alors :

i) Formule de Taylor à l'ordre 1 avec reste intégral

$$f(a + h) = f(a) + \int_0^1 \langle \nabla f(a + th), h \rangle dt.$$

ii) Formule de Taylor-Maclaurin à l'ordre 1

il existe $\theta \in [0, 1]$ tel que $f(a + h) = f(a) + \langle \nabla f(a + \theta h), h \rangle$.

iii) Formule de Taylor-young à l'ordre 1

$$f(a + h) = f(a) + \langle \nabla f(a), h \rangle + o(\|h\|).$$

2. si $f \in C^2(\Omega)$ Alors :

i) Formule de Taylor à l'ordre 2 avec reste intégral

$$f(a + h) = f(a) + \langle \nabla f(a), h \rangle + \int_0^1 (1 - t) \langle \nabla^2 f(a + th)h, h \rangle dt.$$

ii) Formule de Taylor-Maclaurin à l'ordre 2

il existe $\theta \in [0, 1]$ tel que

$$f(a + h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a + \theta h)h, h \rangle.$$

iii) Formule de Taylor-young à l'ordre 2

$$f(a + h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a)h, h \rangle + o(\|h\|^2).$$

Remarque 1.1.2 Dans les formules précédentes, la notation $o(\|h\|^k)$ pour $k \in \mathbb{N}^*$ signifie une expression qui tend vers 0 plus vite que $\|h\|^k$ (c'est à dire, si on la divisé par $\|h\|^k$, le résultat tend vers 0 quand h tend vers 0).

1.2 Programmation mathématique

1.2.1 Définitions

Définition 1.2.1 • Un problème d'optimisation sans contraintes s'écrit comme suit :

$$\begin{cases} \min f(x) = f(\bar{x}) \\ x \in \mathbb{R}^n. \end{cases}$$

Autrement dit :

$$\begin{cases} \text{Trouver } \bar{x} \in \mathbb{R}^n \text{ telle que} \\ f(\bar{x}) \leq f(x), \quad x \in \mathbb{R}^n. \end{cases}$$

• Un problème d'optimisation avec contraintes s'écrit sous la forme :

$$\begin{cases} \min f(x) = f(\bar{x}) \\ x \in S. \end{cases}$$

Autrement dit :

$$\begin{cases} \text{Trouver } \bar{x} \in S \text{ telle que} \\ f(\bar{x}) \leq f(x), \quad x \in S. \end{cases}$$

où $S \subsetneq \mathbb{R}^n$ désigne l'ensemble des contraintes.

Définition 1.2.2 Un problème de programmation mathématique noté (PM) est un problème d'optimisation sous contraintes qui minimise ou maximise une fonction donnée qui peut s'écrire sous la forme suivante :

$$\begin{cases} \min \text{ ou } (\max) f(x) \\ g_i(x) \leq 0, \quad i = \overline{1, n}, \\ h_j(x) = 0, \quad j = \overline{1, m}, \\ x \in S. \end{cases}$$

Où :

1. S est une partie de \mathbb{R}^n et x un vecteur appelé variable, ses n composantes sont dites les inconnues du problème (PM).
2. La fonction $f : S \rightarrow \mathbb{R}$ est appelée fonction objectif ou économique.

3. Les fonctions $g_i : S \rightarrow \mathbb{R}$, $i = 1, \dots, n$, qui forment des inégalités sont appelées les contraintes inégalités du problème.
4. Les fonctions $h_j : S \rightarrow \mathbb{R}$, $j = 1, \dots, m$, qui forment des équations sont appelées les contraintes égalités du problème.
5. Un vecteur x vérifiant les contraintes de (PM), i.e., $g_i(x) \leq 0$, $i = 1, \dots, n$, $h_j(x) = 0$, $j = 1, \dots, m$ et $x \in S$ est dit solution réalisable de (PM), l'ensemble de ces solutions réalisables forme le domaine de définition ou domaine réalisable de (PM), i.e.,

$$D = \text{dom}f \left(\bigcap_{i=1}^n \text{dom}g_i \right) \left(\bigcap_{j=1}^m \text{dom}h_j \right).$$

6. Si $D = \mathbb{R}^n$, On dit que (PM) est un problème d'optimisation sans contraintes.

Définition 1.2.3 (Minimum local)

Soit $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$, on dit que la fonction f admet un minimum local (Solution optimale locale) en $x^* \in D$ si et seulement si :

$$\exists B(x^*, \varepsilon) = \{x \in D, \|x - x^*\| < \varepsilon\} : \quad f(x) \geq f(x^*), \quad \forall x \in B(x^*, \varepsilon).$$

L'ensemble des minima locaux de (PM) et noté par :

$$\text{loc} \min_D f(x).$$

Définition 1.2.4 (Minimum global)

Soit $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$, on dit que la fonction f admet un minimum global (Solution optimale global) en $x^* \in D$ si et seulement si :

$$f(x) \geq f(x^*), \quad \forall x \in D.$$

L'ensemble des minima globaux de (PM) et noté par :

$$\text{arg} \min_D f(x).$$

Remarque 1.2.1 On a toujours $\text{arg} \min_D f(x) \subseteq \text{loc} \min_D f(x)$.

1.2.2 Classification des problèmes

On classifie le problème (PM) à partir de trois propriétés fondamentales à savoir la convexité, la différentiabilité et la linéarité des fonctions du problème.

1. (PM) est convexe si les fonctions f et g_i sont convexes et les fonctions h_j sont affines et D est convexe.
2. (PM) est différentiable si les fonctions f, g_i et h_j sont différentiables.
3. (PM) est linéaire si la fonction f est linéaire et les fonctions g_i et h_j sont affines et D est l'ortant positif.

1.2.3 Principaux résultats d'existence

Théorème 1.2.1 (*Weierstrass*)

Soit D un compact (fermé et borné) non vide de \mathbb{R}^n , si la fonction f est une fonction continue sur D , alors (PM) admet au moins une solution optimale globale $x^* \in D$.

Théorème 1.2.2 Si D est un ensemble convexe et f est strictement convexe, alors il existe au plus une solution optimale de (PM).

Corollaire 1.2.1 Si D est un ensemble non vide et fermé de \mathbb{R}^n , f est une fonction continue et coercive sur D , alors (PM) admet au moins une solution optimale globale $x^* \in D$.

1.2.4 Conditions d'optimalité

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

Une contrainte g_i est dite active (ou saturée) en $\bar{x} \in C$ si $g_i(\bar{x}) = 0$, $i = 1, \dots, n$.

Un point $\bar{x} \in C$ est dit régulier (on dit également que les contraintes sont qualifiées en \bar{x}) si les composantes de gradient, correspondant aux contraintes saturées en \bar{x} , sont linéairement indépendantes.

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 $i = 1, \dots, n$ et qu'il existe un point x^0 tel que $g_i(x^0) < 0$, ($\text{int}(C) \neq \emptyset$).

Définition 1.2.5 *Le lagrangien de programme mathématique (PM) est défini par :*

$$L(x, \lambda, y) = f(x) + \sum_{i=1}^n \lambda_i g_i(x) + \sum_{j=1}^m y_j h_j(x), \lambda_i, y_j \in \mathbb{R}.$$

Théorème 1.2.3 (Karush Kuhn Tucker (KKT)) *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction différentiable sur D , si x^* est un minimum local du problème (PM), alors il existe des vecteurs $y \in \mathbb{R}^m$ et $\lambda \in \mathbb{R}_+^n$ (dits multiplicateurs de Lagrange) tels que :*

$$\nabla f(x^*) + \sum_{i=1}^n \lambda_i \nabla g_i(x^*) + \sum_{j=1}^m y_j \nabla h_j(x^*) = 0, \quad (\text{condition d'optimalité}).$$

$$\lambda_i g_i(x^*) = 0, \quad i = 1, \dots, n, \quad (\text{condition de complémentarité}).$$

$$h_j(x^*) = 0, \quad j = 1, \dots, m.$$

Remarque 1.2.2 *Si (PM) est convexe, alors les conditions d'optimalité (de KKT) sont à la fois nécessaires et suffisantes.*

1.3 Programmation linéaire (PL)

Définition 1.3.1 *Un programme linéaire (PL) est un problème d'optimisation qui consiste à minimiser ou maximiser une fonction objectif linéaire de n variables à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires.*

La représentation mathématique de (PL) est donnée par :

$$(PL) \begin{cases} \text{opt} \left(\sum_{j=1}^n c_j x_j \right) \\ \sum_{j=1}^n a_{ij} x_j (\leq, =, \geq) b_i, \quad i = 1, \dots, m, \\ x_j \geq 0, \end{cases}$$

ou bien sous forme matricielle

$$(PL) \begin{cases} \text{Opt}(c^t x) \\ Ax (\leq, =, \geq) b, \\ x \in \mathbb{R}_+^n. \end{cases}$$

Où (PL) est considéré comme primal, tel que :

- $\text{opt} = \text{max ou min}$.
- $c^t x = \sum_{j=1}^n c_j x_j, \quad c \in \mathbb{R}^n$.
- $a_{ij}, b_i \in \mathbb{R}, \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, n$.

1.3.1 Les formes usuelles d'un programme linéaire

a) La forme canonique

On dit que le problème de programmation linéaire (PL) est sous forme canonique, s'il vérifie les conditions suivantes :

- 1) Toutes ses variables sont non négatives.
- 2) Toutes les contraintes sont de même signe (\geq ou \leq).
- 3) Si la fonction objectif de type de maximisation, alors le signe est (\leq).
- 4) Si la fonction objectif de type de minimisation, alors le signe est (\geq).
- 5) b_i , $i = 1, \dots, m$, sont de signe non déterminé.

Cas de maximisation :

$$\left\{ \begin{array}{l} \max \left(Z = \sum_{j=1}^n c_j x_j \right) \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, m, \\ x_j \geq 0, \quad j = 1, \dots, n. \end{array} \right.$$

Exemple 1.3.1 :

$$\left\{ \begin{array}{l} \max(160x_1 + 210x_2 + 100x_3) \\ 3x_1 + 4x_2 + 2x_3 \leq 57, \\ 2x_1 + x_2 + 2x_3 \leq 2, \\ 4x_1 + 5x_2 + 4x_3 \leq 73, \\ x_1, x_2, x_3 \geq 0. \end{array} \right.$$

Cas de minimisation :

$$\left\{ \begin{array}{l} \min \left(Z = \sum_{j=1}^n c_j x_j \right) \\ \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m, \\ x_j \geq 0, \quad j = 1, \dots, n. \end{array} \right.$$

Exemple 1.3.2 :

$$\left\{ \begin{array}{l} \min(2x_1 + 3x_2 + x_3) \\ 10x_1 + 15x_2 + 10x_3 \geq 20, \\ 100x_1 + 10x_2 + 10x_3 \geq 50, \\ 10x_1 + 100x_2 + 10x_3 \geq 10, \\ x_1, x_2, x_3 \geq 0. \end{array} \right.$$

b) **La forme standard**

On dit que le problème de programmation linéaire (PL) est sous forme standard, s'il vérifie les conditions suivantes :

- 1) Toutes ses variables sont non négatives.
- 2) Toutes les contraintes sous forme des équations.
- 3) La fonction objectif soit maximisé ou minimisé.
- 4) $b_i, i = 1, \dots, m$, sont non négatives.

Cas de maximisation :

$$\left\{ \begin{array}{l} \max \left(Z = \sum_{j=1}^n c_j x_j \right) \\ \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \\ x_j \geq 0, \quad j = 1, \dots, n, \\ b_i \geq 0. \end{array} \right.$$

Exemple 1.3.3 :

$$\begin{cases} \max(30x_1 + 12x_2) \\ 80x_1 + 60x_2 = 42, \\ 12x_1 + 60x_2 = 45, \\ 6x_1 + 2x_2 = 18, \\ x_1, x_2 \geq 0. \end{cases}$$

Cas de minimisation :

$$\begin{cases} \min \left(Z = \sum_{j=1}^n c_j x_j \right) \\ \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \\ x_j \geq 0, \quad j = 1, \dots, n, \\ b_i \geq 0. \end{cases}$$

Exemple 1.3.4 :

$$\begin{cases} \min(5x_1 - 3x_2) \\ x_1 - x_2 - x_3 = 2, \\ 2x_1 + 3x_2 + x_4 = 4, \\ -x_1 + 6x_2 = 10, \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases}$$

c) **La forme mixte**

On dit que le problème de programmation linéaire (PL) est sous forme mixte, s'il n'existe pas **ni** sous forme canonique, **ni** sous forme standard.

Exemple 1.3.5

$$\begin{cases} \max(160x_1 + 21x_2 + 10x_3) \\ 3x_1 + 4x_2 + 2x_3 \leq 57, \\ 2x_1 + 2x_2 + 2x_3 = 2, \\ 4x_1 + 5x_2 + 4x_3 \geq 73, \\ x_1, x_2, x_3 \geq 0. \end{cases}$$

1.3.2 Relation entre forme standard et forme canonique

1. On peut minimiser le problème (PM), comme on peut le maximiser de la manière suivante :

$$\min c^t x = - \max (-c^t x)$$

2. Une contrainte d'égalité $Ax = b$ peut être remplacée par deux inégalités

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

3. Une contrainte d'inégalité peut être transformée en égalité

(a) $Ax \leq b \iff Ax + e = b$, avec $e \geq 0$.

(b) $Ax \geq b \iff Ax - e = b$, avec $e \geq 0$, appelé variable d'écart ou artificielle.

1.3.3 Dualité Lagrangienne

La notion de dualité est un concept fondamental en programmation linéaire qui conduit à un résultat de grande portée théorique et pratique (théorème de dualité faible et théorème de dualité forte). Ainsi, étant donné un problème de programmation linéaire (PL) appelé primal est noté (P), on peut toujours lui associer un autre problème appelé dual (noté par (D)).

Dual d'un programme linéaire

Soit le programme linéaire primal sous forme standard suivant :

$$(P) \quad \{ \min c^t x : Ax = b, x \geq 0, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \}.$$

Lemme 1.3.1 *On appelle dual de (P) le programme linéaire*

$$(D) \quad \{\max b^t y : A^t y \leq c, y \in \mathbb{R}^m\}.$$

preuve . On considère la fonction Lagrangienne suivante

$$\begin{aligned} q(y) &= \min_{x \in \mathbb{R}_+^n} [c^t x + \sum_{i=1}^m y_i (b_i - A_i x), y \in \mathbb{R}^m] \\ &= \min_{x \in \mathbb{R}_+^n} [c^t x + \sum_{i=1}^m b_i y_i - \sum_{i=1}^m y_i A_i x] \\ &= \min_{x \in \mathbb{R}_+^n} [(c^t - \sum_{i=1}^m y_i A_i) x + \sum_{i=1}^m b_i y_i] \\ &= \min_{x \in \mathbb{R}_+^n} [(c^t - A^t y) x + b^t y]. \end{aligned}$$

D'où

$$\max_{y \in \mathbb{R}^m} q(y) = \begin{cases} \max b^t y & \text{si } c^t - A^t y > 0, \\ -\infty & \text{sinon,} \end{cases}$$

donc le problème dual (D) associé au problème primal est :

$$(D) \{\max b^t y : c - A^t y = s, y \in \mathbb{R}^m, s \in \mathbb{R}^n\}.$$

■

Exemple 1.3.6 *Soit le programme linéaire primal sous forme standard :*

$$(P) \quad \{\min(x_1 + x_2) : x_1 - x_2 = 0, x_1 + x_2 + x_3 = 1, x_1, x_2, x_3 \geq 0\}.$$

Sont dual est

$$(D) \quad \{\max y_2 : y_1 + y_2 \leq 1, -y_1 + y_2 \leq 1, y_2 \leq 0\}.$$

1.3.4 Relation primal-dual pour la programmation linéaire

Le Tableau suivant résume les correspondances entre primal et dual et permet d'écrire directement le dual d'un programme linéaire quelconque

Primal	min	max	Dual
	$\geq b_i$	≥ 0	
Contraintes	$\leq b_i$	≤ 0	Variable
	$= b_i$	libre	
Variables	≥ 0	$\geq c_j$	Contraintes
	≤ 0	$\leq c_j$	
	libre	$= c_j$	

Remarque 1.3.1 1. *Le nombre de variables duales est égal au nombre de contraintes primales, de même le nombre de contraintes duales est égal au nombre de variables primales.*

2. *Le dual du programme dual (D) est le programme primal (P).*

3. *On peut transformer un problème de maximisation (respectivement de minimisation) à un problème de minimisation (respectivement de maximisation), il suffit d'écrire $\max(f) = -\min(-f)$ (respectivement $\min(f) = -\max(-f)$).*

Théorème 1.3.1 (dualité faible)

Si x et y sont des solutions réalisables des problèmes (P) et (D) respectivement, alors $c^t x \geq b^t y$.

preuve . Or x et y sont des solutions réalisables, alors

$$c \geq A^t y \Rightarrow c^t x \geq (A^t y)^t x = y^t A x = y^t b = (b^t y)^t = b^t y.$$

Théorème 1.3.2 (dualité forte)

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

preuve On veut démontrer que x^* et y^* sont des solutions optimales des problèmes (P) et (D) respectivement, c'est à dire, on démontre que :

$$c^t x^* = \min c^t x \quad \text{et} \quad b^t y^* = \max b^t y.$$

il vient de montrer que $c^t x \geq b^t y$, donc $\min c^t x \geq \max b^t y$, en particulier

$$b^t y^* = c^t x^* \geq \min c^t x \geq \max b^t y \geq b^t y^* = c^t x^*.$$

Et comme $b^t y^* = c^t x^*$, alors :

$$\begin{cases} c^t x^* \geq \min c^t x \geq c^t x^* \\ \text{et} \\ b^t y^* \geq \max b^t y \geq b^t y^*. \end{cases}$$

d'où le résultat. ■

- Proposition 1.3.1** 1. Si le programme (P) admet une solution optimale x^* , alors le programme (D) admet aussi une solution optimale y^* et celles-ci satisfait $b^t y^* = c^t x^*$.
2. Le programme primal a une solution optimale si et seulement si le programme dual a aussi une.

Pour plus amples de détails sur certains résultats de ce chapitre, le lecteur peut voir par exemple [1, 8, 7, 11, 12].

Remarque 1.3.2 On peut remarquer facilement que si \bar{x} et (\bar{y}, \bar{s}) sont respectivement des solutions réalisables de (P) et (D) , alors on a la propriété suivante :

$$c^t \bar{x} = b^t \bar{y} \iff \bar{x} \bar{s} = 0 \iff \bar{x}^t \bar{s} = 0.$$

1.4 Algorithme d'optimisation

Nous allons présenter un algorithme permettant de converger vers une solution optimale du problème (PM). La plupart des algorithmes d'optimisation avec contraintes exploitent les conditions d'optimalité pour déterminer des minima locaux. Nous donnons ici quelques définitions.

1.4.1 Description

Un algorithme est défini par une application A , de C dans C , où C est l'ensemble des solutions réalisables, permettant la génération d'une suite d'éléments de C par la formule :

$$\begin{cases} x_0 \in C & \text{donné, } k = 0 & \text{Etape d'initialisation} \\ x_{k+1} = A(x_k), k = k + 1 & \text{Itération.} \end{cases}$$

Si on remplace C par son intérieur, en supposant que $\text{int}(C) \neq \emptyset$, l'algorithme est dit un algorithme de points intérieurs.

Définir un algorithme n'est autre que construire une suite $(x_k)_{k \in \mathbb{N}}$ de C et réaliser une étude pour montrer sa convergence.

1.4.2 Convergence

Avant de donner les notations des convergences, nous donnons les définitions des notations asymptotiques suivantes :

Définition 1.4.1 (*Notation \mathcal{O}*)

Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}_+$. On note $f(n) = \mathcal{O}(g(n))$ lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

$$f(n) \leq cg(n).$$

Intuitivement, cela signifie que la valeur de la fonction f est inférieure à celle de g à une constante multiplicative près, pour les instances (données) de tailles suffisamment grandes.

De même on définit :

Définition 1.4.2 (*Notations $\mathbf{o}, \Omega, \Theta$*) Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}_+$.

— On note $f(n) = \mathbf{o}(g(n))$ lorsque pour tout réel c , il existe un entier n_0 tel que pour tout $n \geq n_0$,

$$f(n) \leq cg(n).$$

— On note $f(n) = \Omega(g(n))$ lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

$$cg(n) \leq f(n).$$

— On note $f(n) = \Theta(g(n))$ lorsque $f(n) = \mathcal{O}(g(n))$ et $f(n) = \Omega(g(n))$.

Définition 1.4.3 On dit que l'algorithme A est convergent si la suite $(x_k)_{k \in \mathbb{N}}$ engendrée par l'algorithme converge vers une limite x^* .

1.4.3 Taux de convergence

Un critère de mesure de la vitesse (ou le taux) de convergence est l'évolution de l'erreur commise à chaque itération ($e_k = \|x_k - x^*\|$).

Soit $(x_k)_{k \in \mathbb{N}}$ une suite donnée par l'algorithme A et convergente vers x^* .

La classification de la vitesse de convergence d'une suite est basée sur les notions de comparaison des fonctions au voisinage de $+\infty$.

En effet, si on suppose que l'erreur e_k ne s'annule pas, la vitesse de la convergence pourra être :

Linéaire : Si $\|e_k\| = \Omega(\|e_{k+1}\|)$ et $\left(\frac{\|e_{k+1}\|}{\|e_k\|}\right) < 1$, pour k assez grand. On dit aussi que l'erreur e_k décroît linéairement, c'est à dire :

$$\exists c \in [0, 1[, \quad k_0 \in \mathbb{N}, \quad \forall k \geq k_0, \quad e_{k+1} \leq ce_k.$$

Superlinéaire : Si $\|e_{k+1}\| = \mathbf{o}(\|e_k\|)$, où l'erreur décroît de la manière suivante :

$\exists \alpha_k$ une suite positive qui converge vers 0 tel que $e_{k+1} \leq \alpha_k e_k$.

D'ordre γ avec $\gamma > 1$: Si $\|e_{k+1}\| = \mathcal{O}(\|e_k\|^\gamma)$ et $\left(\frac{\|e_{k+1}\|}{\|e_k\|^\gamma}\right) < 1$, pour k assez grand, où l'erreur décroît de la manière suivante :

$$\exists c \in [0, 1[, \quad \exists k_0 \in \mathbb{N}, \quad \forall k \geq k_0, \quad e_{k+1} \leq c(e_k)^\gamma.$$

Dans le cas $\gamma = 2$, la convergence est dite quadratique.

Chapitre 2

Résolution du problème d'optimisation linéaire via les fonctions noyau

Dans ce chapitre on a choisi de détailler l'article de M. Bouafia et A. Yassine [5], intitulé : " An efficient twice parameterized trigonometric kernel function for linear optimization ". Dans cet article les auteurs ont introduit une nouvelle fonction noyau de type trigonométrique à deux paramètres pour résoudre le problème d'optimisation linéaire, dans le but d'améliorer la complexité algorithmique de MPIs basée sur ce type de fonction noyau.

Rappelons que le problème d'optimisation linéaire (OL), considéré comme primal est défini par

$$(P) \quad \min\{c^t x : Ax = b, x \geq 0\},$$

où $A \in \mathbb{R}^{m \times n}$ avec $\text{rang}(A) = m$; $c \in \mathbb{R}^n$; $b \in \mathbb{R}^m$.

Son problème dual associé est donné par

$$(D) \quad \max\{b^t y : A^t y + s = c, \quad s \geq 0, y \in \mathbb{R}^m\}.$$

Sans perte de généralité, nous supposons que les deux problèmes (P) et (D) satisfont la

condition de points intérieurs (CPI), i.e., il existe (x^0, y^0, s^0) strictement réalisable tel que

$$\begin{cases} Ax^0 = b, \\ A^t y^0 + s^0 = c, \\ x^0, s^0 > 0. \end{cases} \quad (2.1)$$

Il est bien connu que la recherche des solutions optimales de (P) et (D) est équivalente à la résolution du système suivant

$$\begin{cases} Ax = b, \\ A^t y + s = c \\ xs = 0, \\ x \geq 0, \quad s \geq 0. \end{cases} \quad (2.2)$$

L'idée principale des méthodes de points intérieurs (MPIs) de type primal-dual est de remplacer la troisième équation du système (2.2) (dite condition de complémentarité) par l'équation paramétrisée $xs = \mu e$ avec $\mu > 0$ fixé, donc on obtient le système suivant

$$\begin{cases} Ax = b, \\ A^t y + s = c, \\ xs = \mu e, \\ x > 0, \quad s > 0. \end{cases} \quad (2.3)$$

Sous la condition CPI et $\text{rang}(A) = m$, le système (2.3) à une solution unique notée $(x(\mu), y(\mu), s(\mu))$, pour chaque $\mu > 0$.

Nous appelons $x(\mu)$ le μ -centre de (P) et $(y(\mu), s(\mu))$ le μ -centre de (D). L'ensemble des μ -centres construit un chemin d'homotopie, qui est appelé le chemin central de (P) et (D).

Soit $\mu > 0$ fixé, une application directe de la méthode de Newton sur (2.3) conduit au

système linéaire suivant pour les directions Δx , Δy et Δs

$$\begin{cases} A\Delta x = 0, \\ A^t\Delta y + \Delta s = 0 \\ s\Delta x + x\Delta s = \mu e - xs. \\ x > 0, s > 0. \end{cases} \quad (2.4)$$

Ce système admet une solution unique $(\Delta x, \Delta y, \Delta s)$. Nous construisons le nouvel itéré de Newton comme suit

$$(x_+, y_+, s_+) = (x, y, s) + (\Delta x, \Delta y, \Delta s).$$

Pour assurer la stricte réalisabilité de (x_+, y_+, s_+) , on introduit un paramètre $\alpha > 0$, appelé pas de déplacement et dans ce cas l'itéré de Newton modifié s'écrit comme suit :

$$(x_+, y_+, s_+) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s). \quad (2.5)$$

Où le pas de déplacement α satisfait $(0 < \alpha \leq 1)$.

Maintenant, pour passer à la notion des fonctions noyau, nous introduisons le vecteur réduit v et les directions de recherche réduite d_x et d_s comme suit

$$v = \sqrt{\frac{xs}{\mu}}, \quad d_x = \frac{v\Delta x}{x}, \quad d_s = \frac{v\Delta s}{s}. \quad (2.6)$$

Le système (2.4) peut être réécrit comme suit :

$$\begin{cases} \bar{A}d_x = 0, \\ \bar{A}^t\Delta y + d_x = 0, \\ d_x + d_s = v^{-1} - v. \end{cases} \quad (2.7)$$

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

Notons que le côté droit de la troisième équation du système (2.7) est égal au gradient négatif de la fonction barrière logarithmique $\Phi(v)$, *i.e.*,

$$d_x + d_s = -\nabla\Phi(v),$$

dans ce cas le système (2.7) devient

$$\begin{cases} \bar{A}d_x = 0, \\ \bar{A}^t \Delta y + d_s = 0, \\ d_x + d_s = -\nabla \Phi(v). \end{cases} \quad (2.8)$$

Où Φ est définie par

$$\begin{aligned} \Phi(v) : \mathbb{R}_{++}^n &\rightarrow \mathbb{R}_+ \\ v &\mapsto \Phi(v) = \Phi(x, s; \mu) = \sum_{i=1}^n \psi_c(v_i), \end{aligned} \quad (2.9)$$

telle que ψ_c est appelée la fonction noyau classique de la fonction barrière logarithmique Φ , définie par

$$\psi_c(v_i) = \frac{v_i^2 - 1}{2} - \log v_i. \quad (2.10)$$

Tout au long de ce chapitre, nous utilisons la fonction de proximité $\Phi(v)$ pour mesurer la distance entre l'itéré et le μ -centre de la trajectoire centrale pour $\mu > 0$ donné. Nous définissons également la mesure de proximité basée sur la norme, $\delta(v) : \mathbb{R}_{++}^n \rightarrow \mathbb{R}_+$ comme suit

$$\delta(v) = \frac{1}{2} \|\nabla \Phi(v)\| = \frac{1}{2} \|d_x + d_s\|. \quad (2.11)$$

La nouvelle variante des *MPIs* primales-duales a été développée par Peng et al. [9] dont la fonction de proximité $\Phi(v)$ est remplacée par une nouvelle fonction de proximité $\Phi_M(v) = \sum_{i=1}^n \psi_M(v_i)$, qui sera définie dans la section suivante, où $\psi_M(t)$ est une fonction barrière strictement convexe et deux fois différentiable sur \mathbb{R}_{++} avec $\psi_M(1) = \psi'_M(1) = 0$. Maintenant, nous présentons l'algorithme descriptif des *MPIs* basé sur une fonction noyau.

Commençant par le point intérieur (x^0, y^0, s^0) , μ un paramètre positif, $\varepsilon > 0$ une précision donnée et la fonction de proximité $\Phi_M(v)$, soit une bonne approximation du centre $(x(\mu), y(\mu), s(\mu))$, connue pour $\mu > 0$.

En suite, le paramètre μ doit diminuer à un facteur de $(1 - \theta)$, avec $0 < \theta < 1$, et nous

appliquons la méthode de Newton pour trouver les nouveaux μ -centres.

En effet, nous résolvons d'abord le système (2.8) pour trouver d_x et d_s puis utiliser (2.6) pour trouver les directions de Newton $(\Delta x, \Delta y, \Delta s)$. Dans ce cas le nouvel itéré est décrit $(x_+, y_+, s_+) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$.

Cette procédure est répétée jusqu'à ce que nous arrivions au point $x^T s < \varepsilon$ dans ce cas, nous disons que le courant x et (y, s) sont respectivement des solutions ε -approximatives du problème linéaire.

Les paramètres τ , θ et le pas de déplacement α doivent être choisis de manière à ce que l'algorithme est optimisé dans le sens où le nombre d'itérations exigées par l'algorithme est aussi que possible.

Le choix du paramètre barrière de la mise-à-jour θ joue un rôle important dans la théorie et le pratique des MPIs. Si θ est une constante indépendante de la dimension du problème n , par exemple, $\theta = \frac{1}{2}$, alors nous appelons l'algorithme, MPIs à long-pas (ou grande-pas), si θ dépend de la dimension du problème comme $\theta = \frac{1}{n}$, alors l'algorithme est appelé, MPIs à petit-pas (ou court-pas).

La forme générique de cet algorithme est représentée dans la figure1.

Algorithme Primal-dual de *MPIs* pour LO

Début algorithme

Données

Φ_M La fonction de proximité ;

$\tau > 1$, un paramètre de tolérance ;

$\varepsilon > 0$ un paramètre de précision ;

$0 < \theta < 1$; un paramètre barrière fixé ;

Initialisation

$x = e$; $s = e$; $\mu = 1$; $v = e$;

Tant que $x^T s \geq \varepsilon$ **faire**

Début (itération externe)

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

Tant que $\Phi_M(x, s; \mu) > \tau$ **faire**

Début (itération interne)

Résoudre le système (2.8), et utiliser (2.6) pour trouver $(\Delta x, \Delta y, \Delta s);$

Calculer le pas de déplacement $\alpha;$

Poser $x = x + \alpha\Delta x; y = y + \alpha\Delta y; s = s + \alpha\Delta s; v = \sqrt{\frac{xs}{\mu}};$

Fin tant que itération interne)

Fin tant que (itération externe)

Fin algorithme.

Fig 1.

2.1 Propriétés de la nouvelle fonction noyau

Dans cette section, nous présentons certaines propriétés de la fonction noyau paramétrique à terme barrière trigonométrique, qui sont essentielles pour l'analyse de la complexité algorithmique.

Ici, au lieu d'utiliser la fonction noyau classique ψ_c , on utilise la fonction noyau proposée par Bouafia et al. [5] définie comme suit

$$\Psi_M(t) = t^2 + \frac{t^{1-q}}{q-1} - \frac{q}{q-1} + \frac{4}{\pi p} \left[\tan^p h(t) - 1 \right], \quad (2.12)$$

avec

$$\begin{cases} h(t) = \frac{\pi}{2t+2}, \\ p \geq 2, \quad q > 1. \end{cases}$$

Avant d'étudier les propriétés de cette nouvelle fonction noyau, nous avons besoin de donner la définition d'une fonction noyau.

Définition 2.1.1 On appelle $\psi_M : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$ une fonction noyau si elle est deux fois différentiable et vérifié les conditions suivantes :

1. $\psi'_M(1) = \psi_M(1) = 0$.
2. $\psi''_M(t) > 0, \quad \forall t > 0$.
3. $\lim_{t \rightarrow 0^+} \psi_M(t) = \lim_{t \rightarrow \infty} \psi_M(t) = +\infty$.

Les Conditions 1 et 2 montrent que ψ_M est strictement convexe et minimale en $v = e$, avec $\psi_M(e) = 0$.

Ce qui donne

$$\psi_M(e) = 0 \iff \nabla \psi_M(e) = 0 \iff v = e \iff xs = \mu e.$$

Et implique que $\psi_M(t)$ peut s'écrire sous la forme suivante

$$\psi_M(t) = \int_1^t \int_1^\varepsilon \psi''_M(\tau) d\tau d\varepsilon. \quad (2.13)$$

La condition (3) indique que ψ_M est une fonction barrière.

Maintenant, nous donnons les trois premières dérivées de cette fonction noyau.

$$\psi'_M(t) = 2t - t^{-q} + \frac{4}{\pi} \left(1 + \tan^2 h(t) \right) \left(\tan^{p-1} h(t) \right) [h'(t)], \quad (2.14)$$

$$\psi''_M(t) = 2 + qt^{-q-1} + K(t) \left[[(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] (h'(t))^2 + [\tan^{p-1} h(t)] (h''(t)) \right], \quad (2.15)$$

$$\begin{aligned} \psi'''_M(t) = u(t) + K(t) \left[[(p-1)(p-2) \tan^{p-3} h(t) + 2p^2 \tan^{p-1} h(t) + \right. \\ \left. (p+1)(p+2) \tan^{p+1} h(t)] (h'(t))^3 \right. \\ \left. + [3(p-1) \tan^{p-2} h(t) + 3(p+1) \tan^p h(t)] h''(t) \cdot h'(t) \right. \\ \left. + [\tan^{p-1} h(t)] (h'''(t)) \right], \end{aligned} \quad (2.16)$$

avec

$$K(t) = \frac{4}{\pi}(1 + \tan^2 h(t)), \quad (2.17)$$

$$u(t) = -q(q+1)t^{-q-2}, \quad (2.18)$$

et

$$h'(t) = \frac{-\pi}{2(t+1)^2} h''(t) = \frac{\pi}{(t+1)^3} h'''(t) = \frac{-3\pi}{(t+1)^4}. \quad (2.19)$$

Nous présentons dans le lemme suivant quelques propriétés de la fonction h qui sont données dans Bouafia et al.[4].

Lemme 2.1.1 (lemme 3.1 dans Bouafia et al.[4]) Pour $h(t)$ définie dans (2.12) et $p \geq 2$, nous avons le suivant

$$0 < h(t) < \frac{\pi}{2}, \quad t > 0. \quad (2.20)$$

$$\tan h(t) > 0, \quad t > 0. \quad (2.21)$$

$$\tan h(t) - \frac{1}{(p+1)\pi t} > 0, \quad t > 0. \quad (2.22)$$

De (2.5) – (2.15), (2.17), (2.19), et du lemme 2.1.1 et la définition de la fonction noyau dans Bai et al. [3], nous pouvons facilement vérifier que

$$\psi'_M(1) = \psi(1) = 0, \psi''_M(t) > 2.$$

$$\lim_{t \rightarrow 0^+} \psi_M(t) = \lim_{t \rightarrow +\infty} \psi_M(t) = +\infty.$$

Ce qui montre que ψ_M est une fonction noyau.

2.1.1 Éligibilité de la nouvelle fonction noyau

Le lemme suivant sert à prouver que la nouvelle fonction noyau (2.12) est éligible.

Lemme 2.1.2 Soit $\psi_M(t)$, définie dans (2.12) et $t > 0$. Alors

$$\psi'''_M(t) < 0. \quad (2.23)$$

$$t\psi''_M(t) - \psi'_M(t) > 0. \quad (2.24)$$

$$t\psi''_M(t) + \psi'_M(t) > 0. \quad (2.25)$$

preuve.

— Pour (2.23), en utilisant (2.16), (2.17), (2.18), et (2.19) on obtient

$$\psi_M'''(t) < 0, \text{ pour tout } t > 0,$$

ce qui implique (2.23).

— Pour (2.24), on utilise encore (2.14) et (2.15), nous avons

$$\begin{aligned} t\psi_M''(t) - \psi_M'(t) &= 2t + qt^{-q} + tK(t) \left[[(p-1)\tan^{p-2}h(t) + (p+1)\tan^p h(t)](h'(t))^2 \right. \\ &\quad \left. + [\tan^{p-1}h(t)](h''(t)) \right] - 2t + t^{-q} - \frac{4}{\pi}(1 + \tan^2 h(t))(\tan^{p-1}h(t))[h'(t)] \\ &= (q+1)t^{-q} + K(t) \left[t[(p-1)\tan^{p-2}h(t) + (p+1)\tan^p h(t)](h'(t))^2 \right. \\ &\quad \left. + t[\tan^{p-1}h(t)](h''(t)) + (\tan^{p-1}h(t))(-h'(t)) \right]. \end{aligned}$$

De (2.17) et la positivité de $h''(t)$, le coté droit de la dernière équation est positif, ce qui prouve (2.24).

— Pour (2.25), en utilisant (2.14) et (2.15), on a

$$\begin{aligned} t\psi_M''(t) + \psi_M'(t) &= 2t - t^{-q} + \frac{4}{\pi}(1 + \tan^2 h(t))(\tan^{p-1}h(t))[h'(t)] + 2t + qt^{-t} \\ &\quad + tk(t) \left[[(p-1)\tan^{p-2}h(t) + (p+1)\tan^p h(t)](h'(t))^2 + [\tan^{p-1}h(t)](h''(t)) \right] \\ &= 4t + (q-1)t^{-q} + k(t) \left[(h'(t))^2[(p-1)\tan^{p-2}h(t)]t \right. \\ &\quad \left. + [(p+1)\tan^p h(t)](h'(t))^2t + [\tan^{p-1}h(t)](h''(t))t + (\tan^{p-1}h(t))[h'(t)] \right] \\ &= 4t + (q-1)t^{-q} + k(t) \left[(h'(t))^2[(p-1)\tan^{p-2}h(t)]t \right. \\ &\quad \left. + (\tan^{p-1}h(t)) \left((p+1)(h'(t))^2 \tan h(t)t + h''(t)t + h'(t) \right) \right]. \end{aligned}$$

Et

$$((p+1)(h'(t))^2 \tan h(t)t + h''(t)t + h'(t)) = \frac{\pi}{4(t+1)^4} \left(2t^2 + \pi(p+1)t \left[\tan h(t) - \frac{2}{(p+1)\pi t} \right] \right),$$

en utilisant (2.22), le coté droit de l'égalité ci-dessus est positif, ce qui prouve (2.25).

Ce qui termine la preuve. ■

La dernière propriété (2.25) du Lemme 2.1.2 est équivalent à la convexité de la fonction composée $t \mapsto \psi_M(e^t)$ et cela est vrai si et seulement si $\psi_M(\sqrt{t_1 t_2}) \leq \frac{1}{2}(\psi_M(t_1) + \psi_M(t_2))$ pour toutes $t_1, t_2 \geq 0$, cette propriété est connue dans la littérature, elle a été démontrée par plusieurs chercheurs (voir par exemple [9]; [6]; [2]).

Maintenant, nous présentons quelques résultats techniques de la nouvelle fonction noyau.

Lemme 2.1.3 *Pour $\psi_M(t)$, nous avons*

$$(t-1)^2 \leq \psi_M(t) \leq \frac{1}{4}[\psi'_M(t)]^2, \quad t > 0. \quad (2.26)$$

$$\psi_M(t) \leq \left(\frac{3+q}{2} + \frac{\pi}{8}p\right)(t-1)^2, \quad t > 0. \quad (2.27)$$

preuve. Pour montrer (2.26), en utilisant (2.13), on a

$$\begin{aligned} \psi_M(t) &= \int_1^t \int_1^x \psi''_M(y) dy dx \\ &\geq \int_1^t \int_1^x 2 dy dx \\ &= (t-1)^2, \end{aligned}$$

d'une part.

D'autre part, on a

$$\begin{aligned} \psi_M(t) &= \int_1^t \int_1^x \psi''_M(y) dy dx \\ &\leq \frac{1}{2} \int_1^t \int_1^x \psi''_M(y) \psi''_M(x) dy dx \\ &= \frac{1}{2} \int_1^t \psi''_M(x) \left(\int_1^x \psi''_M(y) dy \right) dx \\ &= \frac{1}{2} \int_1^t \left[\psi''_M(x) \psi'_M(y) \Big|_1^x \right] dx \\ &= \frac{1}{2} \int_1^t [\psi''_M(x) (\psi'_M(x) - \psi'_M(1))] dx \\ &= \frac{1}{2} \int_1^t \psi'_M(x) d(\psi'_M(x)) \\ &= \frac{1}{4} (\psi'_M(x))^2, \end{aligned}$$

ce qui donne (2.26).

Pour (2.27), comme $\psi_M(1) = \psi'_M(1) = 0$, $\psi'''_M(1) < 0$, $\psi''_M(1) = (3 + q + \frac{\pi}{4}p)$, et par

l'utilisation de développement de Taylor au voisinage du point 1, on a

$$\begin{aligned}
 \psi_M(t) &= \psi_M(1) + \psi'_M(1)(t-1) + \frac{1}{2}\psi''_M(1)(t-1)^2 + \frac{1}{6}\psi'''_M(\xi)(t-1)^3 \\
 &= \frac{1}{2}\psi''_M(1)(t-1)^2 + \frac{1}{6}\psi'''_M(\xi)(t-1)^3 \\
 &\leq \frac{1}{2}\psi''_M(1)(t-1)^2, \quad \psi'''_M(\xi) < 0 \\
 &= \left(\frac{3+q}{2} + \frac{\pi}{8}p\right)(t-1)^2,
 \end{aligned}$$

pour certains ξ , $1 \leq \xi \leq t$. Ce qui complète la preuve. ■

Soit $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de la fonction noyau $\psi_M(t)$ pour $t \geq 1$ et soit $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de la fonction $-\frac{1}{2}\psi'_M(t)$ pour tout $t \in]0, 1]$.

Alors, nous avons le lemme suivant

Lemme 2.1.4 pour $\psi_M(t)$, nous avons :

$$\sigma(S) \leq 1 + \sqrt{S}, \quad S \geq 0. \quad (2.28)$$

$$\tan h(t) \leq (4Z + 2)^{\frac{1}{p+1}}, \quad Z \geq 0. \quad (2.29)$$

$$t \geq \left[\frac{1}{2Z + 2} \right]^{\frac{1}{q}}, \quad Z \geq 0. \quad (2.30)$$

preuve. Pour (2.28), soit

$$S = \psi_M(t), \quad t \geq 1, \text{ i.e., } \sigma(S) = t, \quad t \geq 1.$$

En utilisant (2.26), on trouve

$$\psi_M(t) \geq (t-1)^2,$$

ce qui implique

$$S \geq (t-1)^2, \quad t \geq 1,$$

d'où

$$t = \sigma(S) \leq 1 + \sqrt{S},$$

ce qui prouve (2.28).

Pour (2.29), soit $Z = -\frac{1}{2}\psi'_M(t)$, $t \in]0, 1]$,

qui est équivalent à $2Z = -\psi'_M(t)$, $t \in]0,1]$,

de la définition de ρ , $\psi'_M(t)$ et $h'(t)$, on a

$$2Z = -\left(2t - t^{-q} + \frac{4}{\pi}(1 + \tan^2 h(t))(\tan^{p-1} h(t))[h'(t)]\right),$$

ce qui donne

$$2Z + 2t - t^{-q} = -\frac{4}{\pi}(1 + \tan^2 h(t))(\tan^{p-1} h(t))(h'(t)),$$

ceci implique que

$$(2Z + 2t - t^{-q})\frac{-\pi}{4h'(t)} = (1 + \tan^2 h(t))(\tan^{p-1} h(t)).$$

Pour $0 < t \leq 1$, on a

$$(2Z + 2t - t^{-q})\frac{-\pi}{4h'(t)} \leq 2(2Z + 1),$$

car $\frac{-1}{t^q} < -1$ et $\frac{\pi}{4}\left(\frac{-1}{h'(t)}\right) \leq \frac{8}{\pi}\frac{\pi}{4} = 2$,

ce qui donne

$$(1 + \tan^2 h(t))(\tan^{p-1} h(t)) \leq 2(2Z + 1),$$

et comme

$$\tan^{p+1} h(t) < (1 + \tan^2 h(t))(\tan^{p-1} h(t)),$$

on trouve

$$\tan h(t) \leq (4Z + 2)^{\frac{1}{p+1}}, \quad Z \geq 0.$$

d'où (2.29).

Pour montrer (2.30), on a

$$2Z + 2t - t^{-q} \geq 0, \quad \forall t \in]0, 1],$$

ce qui implique

$$t^q \geq \frac{1}{2Z + 2}, \quad \forall t \in]0, 1],$$

d'où

$$t \geq \left(\frac{1}{2Z+2}\right)^{\frac{1}{q}}, \forall Z \geq 0,$$

ce qui prouve (2.30).

Ce qui complète la preuve. ■

Dans ce qui suit, nous proposons un théorème important qui est valable pour toute fonction noyau satisfait les conditions (2.23) et (2.24).

Théorème 2.1.1 ([3])

Soit $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de la fonction $\psi_M(t)$ pour $t \geq 1$. Alors, on a

$$\psi_M(\beta v) \leq n\psi_M\left(\beta\sigma\left(\frac{\Phi_M(v)}{n}\right)\right), \quad v \in \mathbb{R}_{++}^n, \beta \geq 1.$$

Lemme 2.1.5 Soit $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$, si $\Phi_M(v) \leq \tau$, Alors on a

$$\Phi_M(v_+) \leq \frac{\theta n + \tau + 2\sqrt{\tau n}}{1-\theta}.$$

preuve. Comme le vecteur v est positif et $\frac{1}{\sqrt{1-\theta}} \geq 1$, et $\sigma\left(\frac{\Phi_M(v)}{n}\right) \geq 1$, alors

$$\frac{\sigma\left(\frac{\Phi_M(v)}{n}\right)}{\sqrt{1-\theta}} \geq 1,$$

et pour tout $t \geq 1$, on a

$$\psi_M(t) \leq t^2 - 1.$$

En utilisant le théorème 2.1.1 avec $\beta = \frac{1}{\sqrt{1-\theta}}$, pour $\Phi_M(v) \leq \tau$, on trouve

$$\begin{aligned} \Phi_M(v_+) &= \Phi_M(\beta v) \\ &\leq n\psi_M\left(\beta\sigma\left(\frac{\Phi_M(v)}{n}\right)\right) \\ &= n\psi_M\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_M(v)}{n}\right)\right) \\ &\leq n\left(\left[\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_M(v)}{n}\right)\right]^2 - 1\right) \\ &= \frac{n}{1-\theta}\left(\left[\sigma\left(\frac{\Phi_M(v)}{n}\right)\right]^2 - (1-\theta)\right). \end{aligned}$$

De (2.28), on a

$$\begin{aligned}
 \Phi_M(\beta v) &\leq \frac{n}{1-\theta} \left(\left[1 + \sqrt{\left(\frac{\Phi_M(v)}{n}\right)} \right]^2 + \theta - 1 \right) \\
 &= \frac{n}{1-\theta} \left(\left[1 + \left(\frac{\Phi_M(v)}{n}\right) + 2\sqrt{\left(\frac{\Phi_M(v)}{n}\right)} \right] + \theta - 1 \right) \\
 &\leq \frac{n}{1-\theta} \left(\theta + \frac{\tau}{n} + 2\sqrt{\frac{\tau}{n}} \right) \\
 &= \frac{n\theta + \tau + 2\sqrt{n\tau}}{1-\theta}.
 \end{aligned}$$

Ce qui termine la preuve. ■

On note par

$$(\Phi_M)_0 = \frac{\theta n + \tau + 2\sqrt{n\tau}}{1-\theta} = L(n, \theta, \tau). \quad (2.31)$$

Alors, $\Phi_M(v_+)$ est majoré par $(\Phi_M)_0$ dans notre algorithme.

2.1.2 Détermination du pas de déplacement

Dans cette partie, on va calculer le pas de déplacement et prouver que la fonction de proximité Φ est décroissante à chaque itération interne puis on va donner les résultats de la complexité algorithmique.

Pour μ fixé si on prend α le pas de déplacement, alors le nouvel itéré est donné par

$$(x_+, y_+, s_+) := (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s).$$

En utilisant (2.6), on trouve

$$\begin{aligned}
 x_+ &:= x + \alpha\Delta x = x\left(e + \alpha\frac{\Delta x}{x}\right) = x\left(e + \alpha\frac{d_x}{v}\right) = \frac{x}{v}(v + \alpha d_x) \\
 s_+ &:= s + \alpha\Delta s = s\left(e + \alpha\frac{\Delta s}{s}\right) = s\left(e + \alpha\frac{d_s}{v}\right) = \frac{s}{v}(v + \alpha d_s).
 \end{aligned}$$

Donc, nous avons

$$v_+ = \sqrt{\frac{x_+ s_+}{\mu}} = \sqrt{(v + \alpha d_x)(v + \alpha d_s)}.$$

Posons $f(\alpha) = \Phi_M(v_+) - \Phi_M(v)$, pour $\alpha > 0$.

Alors $f(\alpha)$ est la différence de la proximité entre le nouvel et l'ancien itéré, pour μ fixé.

d'après (2.25) du lemme 2.1.2, on a

$$\Phi_M(v_+) = \Phi_M\left(\sqrt{(v + \alpha dx)(v + \alpha ds)}\right) \leq \frac{\Phi_M(v + \alpha dx) + \Phi_M(v + \alpha ds)}{2},$$

ce qui implique

$$f(\alpha) \leq \frac{\Phi_M(v + \alpha dx) + \Phi_M(v + \alpha ds)}{2} - \Phi_M(v) =: f_1(\alpha). \quad (2.32)$$

Il est clair que $f(0) = f_1(0) = 0$.

Prenant la première dérivée de $f_1(\alpha)$ par rapport à α , nous avons

$$f_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi_M'(v_i + \alpha dx_i) dx_i + \psi_M'(v_i + \alpha ds_i) ds_i),$$

où dx_i et ds_i désigne respectivement la i^{eme} composante des vecteurs dx et ds .

En utilisant (2.8) et (2.11), on trouve

$$\begin{aligned} f_1'(0) &= \frac{1}{2} \sum_{i=1}^n \left[\psi_M'(v_i) dx_i + \psi_M'(v_i) ds_i \right] \\ &= \frac{1}{2} \sum_{i=1}^n \psi_M'(v_i) (dx_i + ds_i) \\ &= \frac{1}{2} \nabla \Phi_M^t(v) (dx + ds) \\ &= -\frac{1}{2} \nabla \Phi_M^t(v) \nabla \Phi_M(v) \\ &= -2 \left(\delta(v) \right)^2. \end{aligned}$$

La deuxième dérivée de $f_1(\alpha)$ par rapport à α est donnée par

$$f_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi_M''(v_i + \alpha dx_i) d^2 x_i + \psi_M''(v_i + \alpha ds_i) d^2 s_i).$$

Lemme 2.1.6 *Soit $\delta(v)$ définie dans (2.11). Alors, nous avons*

$$\delta(v) \geq \sqrt{\Phi_M(v)}. \quad (2.33)$$

preuve. D'après (2.26), on a :

$$\psi_M(t) \leq \frac{1}{4} [\psi_M'(t)]^2,$$

ce qui implique

$$\begin{aligned}\Phi_M(v) &= \sum_{i=1}^n \psi_M(v_i) \leq \sum_{i=1}^n \frac{1}{4} (\psi'_M(v_i))^2 \\ &= \frac{1}{4} \sum_{i=1}^n (\psi'_M(v_i))^2 = \frac{1}{4} \|\nabla \Phi_M(v)\|^2 \\ &= (\delta(v))^2,\end{aligned}$$

ce qui termine la preuve. ■

Remarque 2.1.1 *Tout au long de cette partie, nous supposons que $\tau \geq 1$. En utilisant le lemme 2.1.6 et l'hypothèse $\Phi_M(v) \geq \tau$, nous obtenons que*

$$\delta(v) \geq 1.$$

D'après les lemmes 4.1-4.4 dans Bai et al. [3], nous avons les lemmes 2.1.7-2.1.10 suivants

Lemme 2.1.7 *Soit $f_1(\alpha)$ définie dans (2.32) et $\delta(v)$ définie dans (2.11), alors on a :*

$$f_1''(\alpha) \leq 2\delta^2 \psi''_M(v_{\min} - 2\alpha\delta).$$

Lemme 2.1.8 *Si le pas de déplacement α vérifie l'inégalité suivante*

$$\psi'_M(v_{\min}) - \psi'_M(v_{\min} - 2\alpha\delta) \leq 2\delta, \tag{2.34}$$

alors :

$$f_1'(\alpha) \leq 0.$$

Lemme 2.1.9 *Soit $\rho : [0, +\infty[\rightarrow]0, 1]$ l'inverse de la fonction $\frac{-1}{2}\psi'_M(t)$ pour tout $t \in]0, 1]$. Alors, la valeur maximale du pas de déplacement $\bar{\alpha}$ qui vérifie l'inégalité*

$$\psi'_M(v_{\min}) - \psi'(v_{\min} - 2\alpha\delta) \leq 2\delta.$$

est donnée par

$$\bar{\alpha} = \frac{\rho(\delta) - \rho(2\delta)}{2\delta}.$$

Lemme 2.1.10 *Soit $\bar{\alpha}$ définie dans le lemme 2.1.9. Alors*

$$\bar{\alpha} \geq \frac{1}{\psi''_M(\rho(2\delta))}.$$

Lemme 2.1.11 Soient ρ et $\bar{\alpha}$ définient dans le lemme 2.1.10. Si $\Phi_M(v) \geq \tau \geq 1$, alors on a

$$\bar{\alpha} \geq \frac{1}{(4p\pi + 11)(8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}}}.$$

preuve. D'après le lemme 2.1.10, on a

$$\bar{\alpha} \geq \frac{1}{\psi''_M(\rho(2\delta))}$$

et pour $\rho : \rho(Z) = t$, $t \in]0, 1]$ et d'après (2.19), nous avons

$$h'(t) = \frac{-\pi}{2(t+1)^2} \Rightarrow (h'(t))^2 = \frac{\pi^2}{4(t+1)^4} \leq \left(\frac{\pi}{2}\right)^2,$$

$$h''(t) = \frac{\pi}{(t+1)^3} \leq \pi.$$

De (2.15), on a

$$\begin{aligned} \psi''_M(t) &= \frac{4}{\pi}(1 + \tan^2 h(t)) \left[\left[(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t) \right] (h'(t))^2 \right. \\ &\quad \left. + \left[\tan^{p-1} h(t) \right] (h''(t)) \right] + 2 + qt^{-q-1}. \end{aligned}$$

Pour $Z = 2\delta$ et de (2.29), on a $\tan h(t) \leq (4Z + 2)^{\frac{1}{p+1}}$, ce qui implique

$$\tan^2 h(\rho(2\delta)) \leq (8\delta + 2)^{\frac{2}{p+1}}, \quad (2.35)$$

$$\tan^{p-2} h(\rho(2\delta)) \leq (8\delta + 2)^{\frac{p-2}{p+1}}, \quad (2.36)$$

$$\tan^{p-1} h(\rho(2\delta)) \leq (8\delta + 2)^{\frac{p-1}{p+1}} \quad (2.37)$$

et

$$\tan^p h(\rho(2\delta)) \leq (8\delta + 2)^{\frac{p}{p+1}}. \quad (2.38)$$

De (2.30), on a

$$t \geq \left(\frac{1}{2Z + 2} \right)^{\frac{1}{q}}$$

et comme $8\delta + 1 > 1$, on trouve

$$t \geq \left(\frac{1}{8\delta + 2} \right)^{\frac{1}{q}} \quad (2.39)$$

on a $8\delta + 2 > 1$, ce qui implique

$$\begin{aligned}\psi''_M(t) &\leq 2 + (8\delta + 2)^{\frac{q+1}{q}} + \frac{4}{\pi} \left(2 \right) \left[\left((2p)^{\frac{\pi^2}{4}} + \pi \right) (8\delta + 2)^{\frac{p+2}{p+1}} \right] \\ &\leq (4p\pi + 11)(8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}},\end{aligned}$$

d'après le lemme 2.1.10, on a

$$\bar{\alpha} \geq \frac{1}{\psi''_M(\rho(2\delta))},$$

ce qui donne

$$\bar{\alpha} \geq \frac{1}{(4p\pi + 11)(8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}}}.$$

On note par

$$\tilde{\alpha} = \frac{1}{(4p\pi + 11)(8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}}}. \quad (2.40)$$

Où $\tilde{\alpha}$ est le pas de déplacement par défaut et on a $\tilde{\alpha} \leq \bar{\alpha}$.

Lemme 2.1.12 (*Lemme 4.5 dans [3]*) Si le pas de déplacement α vérifie $\alpha \leq \bar{\alpha}$, alors

$$f(\alpha) \leq -\alpha\delta^2.$$

Lemme 2.1.13 Soient $\tilde{\alpha}$ le pas de déplacement par défaut définie dans (2.40) et $\Phi_M(v) \geq 1$, alors

$$f(\tilde{\alpha}) \leq -\frac{1}{1400(p+1)} \left[(\Phi_M)_0 \right]^{\frac{pq-p-2}{2(p+1)q}}. \quad (2.41)$$

preuve . En utilisant le lemme 2.1.12 avec $\alpha = \tilde{\alpha}$ et (2.40), nous avons

$$f(\tilde{\alpha}) \leq -\tilde{\alpha}\delta^2 = -\frac{\delta^2}{(4p\pi + 11)(8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}}},$$

comme

$$(8\delta + 2) \leq (8\delta + 2\delta) = 10\delta, \quad \delta > 1,$$

ce qui implique

$$10^2 \delta^{\frac{(p+2)(q+1)}{(p+1)q}} \leq (8\delta + 2)^{\frac{(p+2)(q+1)}{(p+1)q}} \leq 10^3 \delta^{\frac{(p+2)(q+1)}{(p+1)q}},$$

d'où

$$\begin{aligned}
 f(\tilde{\alpha}) &\leq -\frac{\delta^2}{100(4p\pi + 11)\delta^{\frac{(p+2)(q+1)}{(p+1)q}}} \\
 &= -\frac{\delta^{2-\frac{(p+2)(q+1)}{(p+1)q}}}{100(4p\pi + 11)} \\
 &\leq -\frac{\delta^{\frac{pq-p-2}{(p+1)q}}}{100(14p + 11)} \\
 &\leq -\frac{\delta^{\frac{pq-p-2}{(p+1)q}}}{1400(p + 1)} \\
 &\leq -\frac{1}{1400(p + 1)} \left[(\Phi_M)(v) \right]^{\frac{pq-p-2}{2(p+1)q}} \\
 &\leq -\frac{1}{1400(p + 1)} \left[(\Phi_M)_0 \right]^{\frac{pq-p-2}{2(p+1)q}},
 \end{aligned}$$

puisque

$$\delta(v) \geq \sqrt{\Phi_M(v)} \geq 1,$$

ce qui termine la preuve. ■

2.2 Nombre d'itérations de l'algorithme

2.2.1 Nombre d'itérations internes

Après la mise à jour de μ vers $(1 - \theta)\mu$, nous avons

$$\Phi_M(v_+) \leq (\Phi_M)_0 = \frac{\theta n + \tau + 2\sqrt{\tau n}}{1 - \theta} = L(n, \theta, \tau).$$

Nous devons compter le nombre d'itérations internes nécessaires pour revenir à la situation

$$\Phi_M(v) \leq \tau.$$

Nous notons par $(\Phi_M)_0$, la valeur de $\Phi_M(v)$ après une μ mise à jour ; les valeurs des suites dans la même itération externe sont dénotées par $(\Phi_M)_k$, $k = 1, 2, \dots, K$, où K désigne le nombre total d'itérations internes dans une itération externe. Le décroissement de la fonction de proximité (Φ_M) dans chaque itération interne est garantis et donné par la

relation (2.41). Dans [3], nous pouvons trouver la valeur appropriée de \bar{k} et $\gamma \in]0, 1]$

$$\bar{k} = \frac{1}{1400(p+1)}, \quad \gamma = 1 - \frac{pq - p - 2}{2(p+1)q} = \frac{(p+2)(q+1)}{2(p+1)q}.$$

Lemme 2.2.1 *Soit K le nombre total d'itérations internes dans une itération externe. Alors, nous avons*

$$K \leq \left(\frac{2800(p+1)^2q}{(p+2)(q+1)} \right) \left[(\Phi_M)_0 \right]^{\frac{(p+2)(q+1)}{2(p+1)q}}.$$

Preuve. Pour la démonstration de ce lemme, nous avons besoin du lemme 1.3.2 de Peng et al. [9] suivant

Lemme 2.2.2 *Soit t_0, t_1, \dots, t_k une suite des nombres positifs qui vérifie*

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

tels que $\beta > 0$ et $0 < \gamma \leq 1$, alors :

$$K \leq \left\lceil \frac{t_0^\gamma}{\beta\gamma} \right\rceil.$$

Posons

$$t_K = \Phi_M, \quad \beta = \frac{1}{1400(p+1)}, \quad \text{et } \gamma = 1 - \frac{pq - p - 2}{2(p+1)q} = \frac{(p+2)(q+1)}{2(p+1)q},$$

donc on trouve

$$K \leq \left(\frac{2800(p+1)^2q}{(p+2)(q+1)} \right) \left[(\Phi_M)_0 \right]^{\frac{(p+1)(q+1)}{2(p+1)q}},$$

ce qui termine la preuve. ■

2.2.2 Nombre total d'itérations

Le nombre d'itération externes est borné ci-dessus par $\frac{\log \frac{n}{\epsilon}}{\theta}$ (voir Roos et al. dans [?], lemme II.17 page 116).

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

$$\left(\frac{2800(p+1)^2q}{(p+2)(q+1)} \right) \left[(\Phi_M)_0 \right]^{\frac{(p+2)(q+1)}{2(p+1)q}} \frac{\log \frac{n}{\epsilon}}{\theta}. \quad (2.42)$$

Pour les méthodes à long-pas, avec $\tau = \mathcal{O}(n)$ et $\theta = \Theta(\mathbf{1})$, on a

$$\mathcal{O}\left(pn^{\frac{(p+2)(q+1)}{2(p+1)q}} \log \frac{n}{\epsilon}\right) \text{ itérations de complexité.}$$

Remarque 2.2.1 *On voit que si p et q sont des paramètres dépendants de n , en particulier $p = q = \log n$, nous obtenons la meilleure complexité pour les méthodes à long-pas, à savoir $\mathcal{O}\left(\sqrt{n}(\log n) \log \frac{n}{\epsilon}\right)$.*

Dans le cas des méthodes à court-pas, on a $\tau = \mathcal{O}(1)$ et $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$.

Le remplacement de ces valeurs dans (2.42) ne donne pas la meilleure borne possible.

Une meilleure borne est obtenue comme suit : En utilisant (2.27) avec

$$\psi_M(t) \leq \left(\frac{3+q}{2} + \frac{\pi p}{8}\right)(t-1)^2, \quad t > 1,$$

on a

$$\begin{aligned} \Phi_M(v_+) &\leq n\psi_M\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_M(v)}{n}\right)\right) \\ &\leq n\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_M(v)}{n}\right) - 1\right)^2 \\ &= \frac{n\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)}{(1-\theta)}\left(\sigma\left(\frac{\Phi_M(v)}{n}\right) - \sqrt{1-\theta}\right)^2 \\ &\leq \frac{n\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)}{(1-\theta)}\left(\left(1 + \sqrt{\frac{\Phi_M(v)}{n}}\right) - \sqrt{1-\theta}\right)^2 \\ &= \frac{n\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)}{(1-\theta)}\left(\left(1 - \sqrt{1-\theta}\right) + \sqrt{\frac{\Phi_M(v)}{n}}\right)^2 \\ &\leq \frac{n\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)}{(1-\theta)}\left(\theta + \sqrt{\frac{\tau}{n}}\right)^2 \\ &= \frac{\left(\frac{3+q}{2} + \frac{\pi p}{8}\right)}{(1-\theta)}\left(\theta\sqrt{n} + \sqrt{\tau}\right)^2 = (\Phi_M)_0, \end{aligned}$$

où nous avons également utilisé la propriété $1 - \sqrt{1-\theta} = \frac{\theta}{1+\sqrt{1-\theta}} \leq \theta$ et $\Phi_M(v) \leq \tau$.

En utilisant cette borne supérieure $(\Phi_M)_0$, nous obtenons le majorant de nombre total

d'itérations suivant

$$\left(\frac{2800(p+1)^2q}{(p+2)(q+1)}\right) \left[(\Phi_M)_0\right]^{\frac{(p+2)(q+1)}{2(p+1)q}} \frac{\log \frac{n}{\epsilon}}{\theta}.$$

Remarque 2.2.2 Si $(\Phi_M)_0 = \mathcal{O}(p+q)$, la complexité algorithmique pour les méthodes à court-pas devient

$$\mathcal{O}\left((p+q)^2 \sqrt{n} \log \frac{n}{\epsilon}\right) \text{ itérations.}$$

Et la meilleure complexité $\mathcal{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$ est obtenue si on prend tout simplement $p+q = \mathcal{O}(1)$.

Chapitre 3

Tests numériques

Dans ce chapitre, on présente les résultats numériques d'un test à taille variable effectué sur l'algorithme étudié dans le chapitre 2, on prenant en considération tous les aménagements effectués.

L'exemple est pris de la littérature (voir par exemple [12]) et implémenté dans MATLAB R2008b sur Intel Core i3 (1.80 GHz) avec 4.00 Go RAM. On a pris $\epsilon = 1e - 8$, $\tau = \sqrt{n}$ et $\theta \in \{0.01, 0.1, 0.5, 0.95, 0.99\}$.

Dans l'exemple testé nous disposons d'une solution initiale primale-duale strictement réalisable (x^0, y^0, s^0) . Dans les tableaux des résultats, $(\text{par}(p, q))$ représente les valeurs des paramètres p et q , $(\text{taille}(m, n))$ représente la taille de l'exemple, (Itr) représente le nombre des itérations externes (internes) nécessaire pour obtenir une solution optimale, (temps) représente le temps de calcul et (saut) représente le saut de dualité (ou le test d'arrêt).

Rappelons que le problème primal considéré est

$$(P) \quad \min\{c^t x : Ax = b, x \geq 0\},$$

où son problème dual associé est :

$$(D) \quad \max\{b^t y : A^t y + s = c, \quad s \geq 0\}.$$

L'exemple qu'on a implémenté est un exemple à taille variable définit comme suit

$$n = 2m, A(i, j) = \begin{cases} 0 & \text{si } i \neq j \text{ et } j \neq i + m, \\ 1 & \text{si } i = j \text{ ou } j = i + m, \end{cases}$$

$$c(i) = c(i + m) = -1, \quad b(i) = 2, \text{ pour } i = 1, \dots, m.$$

La solution primale-duale strictement réalisable initiale prise est :

$$x^0(i) = 1.5, \quad x^0(i + m) = 0.5, \quad y^0(i) = -2 \text{ et } s^0(i) = s^0(i + m) = 1, \text{ pour } i = 1, \dots, m.$$

La solution optimale primale-duale trouvée pour $m \leq 25$ est :

$$x^*(i) = 1, \quad y^*(i) = -1 \text{ et } s^*(i) = 0, \text{ pour } i = 1, \dots, m.$$

La valeur optimale pour $m \leq 25$ est :

$$c^t x^* = b^t y^* = -n = -2m.$$

3.1 Choix des paramètres p et q

Les tableaux suivants résument les résultats obtenus dans notre algorithme qui concernent le nombre d'itérations, le temps de calcul et le saut de dualité pour chaque choix de paramètres p et q en prenant en considération les différentes tailles du problèmes.

Table 1 : Taille $n = 2m = 10$.

par (p,q)		$\theta = 0.99$	$\theta = 0.95$	$\theta = 0.5$	$\theta = 0.1$	$\theta = 0.05$	$\theta = 0.01$
$p = e^{-n}$ $q = \ln n$	Iter	19(5)	16(7)	29(29)	190(190)	390(390)	1990(1990)
	temps	0.3443	0.3207	0.4240	1.2761	2.3192	10.9154
	saut	1.3972e-9	8.5481e-9	9.3132e-9	9.3132e-9	9.3132e-9	9.3132e-9
$p = e^{-n}$ $q = e^{-n}$	Iter	18(5)	17(8)	31(31)	199(199)	407(407)	2077(2077)
	temps	0.3562	0.2893	0.4221	1.3378	2.4163	11.3878
	saut	3.6029e-9	1.4838e-9	9.3132e-9	9.3132e-9	9.3132e-9	9.3132e-9
p=ln(n) q=ln(n)	Iter	23(5)	27(7)	52(31)	251(201)	469(416)	2184(2127)
	temps	0.3997	0.461	0.7933	2.1703	3.3439	12.647
	saut	1.5271e-9	8.3514e-9	9.3676e-9	9.7423e-9	8.3808e-9	8.2450e-9
p=ln(n) q=6	Iter	23(5)	32(8)	54(32)	235(210)	455(430)	2162(2137)
	temps	0.4028	0.5433	0.8684	1.883	2.9963	12.1149
	saut	2.4865e-9	1.6169e-9	4.4987e-9	4.5357e-9	4.7428e-9	8.1063e-9
p=2 q=6	Iter	24(5)	34(8)	56(32)	232(207)	448(423)	2181(2156)
	temps	0.4089	0.5730	0.8731	2.3742	3.6665	12.3521
	saut	2.9409e-9	7.539e-10	5.182e-9	5.5907e-9	5.7836e-9	5.9371e-9
p=4 q=6	Iter	19(5)	22(8)	45(33)	235(211)	453(429)	2187(2163)
	temps	0.3411	0.4183	0.6661	1.8251	2.9897	12.0931
	saut	2.2119e-9	9.5413e-10	3.5800e-9	6.1296e-9	7.583e-9	9.8136e-9
p=4 q=ln(n)	Iter	22(5)	20(8)	65(31)	234(213)	447(426)	2181(2160)
	temps	0.3801	0.3792	0.8955	1.8301	2.9049	12.0937
	saut	3.3520e-9	1.1099e-9	8.1129e-9	4.0244e-9	7.2947e-9	8.4022e-9

Table 2 : Taille $n = 2m = 20$.

par (p,q)		$\theta = 0.99$	$\theta = 0.95$	$\theta = 0.5$	$\theta = 0.1$	$\theta = 0.05$	$\theta = 0.01$
$p = e^{-n}$ $q = \ln n$	Iter	17(5)	17(8)	31(31)	204(204)	419(419)	2137(2137)
	temps	2.9439	2.3529	2.2183	2.9862	3.7313	8.4996
	saut	2.3305e-9	1.1642e-9	4.6566e-9	4.6566 e-9	4.6566e-9	4.6566e-9
$p = e^{-n}$ $q = e^{-n}$	Iter	18(5)	17(8)	33(33)	211(211)	434(434)	2212(2212)
	temps	2.2831	2.2015	2.2672	2.8983	3.5375	8.5996
	saut	7.2029e-9	2.9670e-9	4.6566e-9	4.6566e-9	4.6566e-9	4.6566e-9
$p = \ln n$ $q = \ln n$	Iter	23(5)	26(8)	47(32)	240(216)	463(439)	2243(2219)
	temps	2.7525	3.0449	3.7220	5.3653	5.9920	10.5212
	saut	4.8214e-9	1.3905e-9	9.9371e-9	4.9484e-9	6.3573e-9	7.7883e-9
$p = \ln n$ $q = 6$	Iter	24(5)	36(8)	53(33)	238 (213)	460 (435)	2297 (2270)
	temps	2.7912	3.8318	4.3625	5.4975	6.1966	11.0339
	saut	4.1945e-9	1.8444e-9	5.4850e-9	7.7173e-9	8.9142e-9	5.2928e-9
p=2 q=6	Iter	24(5)	35(8)	58(33)	240(214)	464(438)	2257(2231)
	temps	2.9104	4.1403	4.9825	5.9726	6.5470	10.7105
	saut	5.8817e-9	2.4830e-10	5.7517e-9	5.0090e-9	5.1943e-9	5.3586e-9
p=4 q=6	Iter	19(5)	22(8)	45(33)	244(219)	469(444)	2271(2246)
	temps	2.3221	2.5604	3.5155	5.8368	6.3031	11.0539
	saut	4.4239e-9	1.9083e-9	7.1600e-9	5.2772e-9	6.9385e-9	8.4909e-9
$p = 4$ $q = \ln n$	Iter	26(6)	24(8)	55(32)	236(214)	467(445)	2238(2216)
	temps	2.9602	3.0089	4.0645	5.1572	5.6656	11.1133
	saut	4.4910e-11	4.1316e-9	7.8505e-9	7.5748e-9	5.8056e-9	9.9732e-9

Table 3 : Taille $n = 2m = 30$.

par (p,q)		$\theta = 0.99$	$\theta = 0.95$	$\theta = 0.5$	$\theta = 0.1$	$\theta = 0.05$	$\theta = 0.01$
$p = e^{-n}$ $q = \ln n$	Iter	17(5)	17(8)	32(32)	204(204)	419(419)	2139(2139)
	temps	10.8530	10.5646	10.1934	10.9104	11.2234	16.1372
	saut	3.1304e-9	2.1836e-9	6.9849e-9	6.9849e-9	6.9849e-9	6.9849e-9
$p = e^{-n}$ $q = e^{-n}$	Iter	22(6)	17(8)	33(33)	211(211)	434(434)	2211(2211)
	temps	13.5598	10.7646	10.4003	11.0664	11.2104	16.3506
	saut	1.1709e-10	4.4505e-9	6.9849e-9	6.9849e-9	6.9849e-9	6.9849e-9
$p = \ln n$ $q = \ln n$	Iter	28(6)	22(8)	50(34)	245(221)	472(448)	2286(2262)
	temps	16.4539	13.2613	19.2038	27.6431	28.2404	33.5015
	saut	7.0253e-11	3.2870e-9	3.8346e-9	5.1731e-9	6.8855e-9	8.5299e-9
$p = \ln n$ $q = 6$	Iter	33(6)	34(8)	53(33)	243 (217)	485 (458)	2352 (2325)
	temps	19.4342	19.6359	23.6999	29.7539	31.8961	36.6617
	saut	1.4873e-10	2.7834e-9	8.3264e-9	8.4908e-9	4.4687e-9	5.0656e-9
$p = 2$ $q = 6$	Iter	24(5)	35(8)	60(34)	239(213)	463(437)	2253(2227)
	temps	14.3759	20.3767	29.6440	29.8478	30.3551	35.2400
	saut	8.8226e-9	3.7245e-9	4.1932e-9	7.8039e-9	8.0391e-9	8.2275e-9
$p = 4$ $q = 6$	Iter	19(5)	22(8)	45(33)	244(219)	487(461)	2357(2331)
	temps	11.8356	13.3396	18.5085	28.7652	30.8297	35.5451
	saut	6.6358e-9	9.8624e-9	9.2862e-9	7.9158e-9	4.3832e-9	5.3626e-9
$p = 4$ $q = \ln n$	Iter	26(6)	23(8)	57(33)	246(223)	475(452)	2303(2280)
	temps	16.1338	13.8638	23.1054	26.7125	27.2168	31.2266
	saut	9.9448e-11	6.9855e-9	8.5044e-9	4.5518e-9	6.1158e-9	8.0211e-9

Table 4 : Taille $n = 2m = 40$.

par (p,q)		$\theta = 0.99$	$\theta = 0.95$	$\theta = 0.5$	$\theta = 0.1$	$\theta = 0.05$	$\theta = 0.01$
$p = e^{-n}$ $q = \ln n$	Iter	17(5)	19(8)	32(32)	205(205)	420(420)	2139(2139)
	temps	43.0737	48.0840	41.6514	41.8870	42.9443	47.2440
	saut	3.7946e-9	1.9071e-9	9.3132e-9	9.3132e-9	9.3132e-9	9.3132e-9
$p = e^{-n}$ $q = e^{-n}$	Iter	22(6)	17(8)	32(32)	211(211)	434(434)	2210(2210)
	temps	61.0717	58.1170	45.9872	48.3399	(2eme fois	60.0303
	saut	1.5612e-10	5.9339e-9	9.3132e-9	9.3132e-9	62.1642)	9.3132e-9
$p = \ln n$ $q = \ln n$	Iter	22(6)	21(8)	52(35)	249(225)	478(454)	2312(2288)
	temps	55.2538	76.2905	89.0824	135.7325	151.1697	146.8774
	saut	2.3242e-10	4.7177e-9	2.6166e-9	4.8111e-9	7.2050e-9	9.4742e-9
$p = \ln n$ $q = 6$	Iter	25(6)	25(8)	51(34)	248 (222)	477 (451)	2392 (2365)
	temps	65.9262	67.4164	92.1990	134.1217	141.5299	158.3072
	saut	1.4875e-10	9.3585e-9	4.2495e-9	7.4369e-9	9.3044e-9	4.7719e-9
$p = 2$ $q = 6$	Iter	31(6)	37(8)(37(8))	59(33)	248(221)	480(453)	2335(2308)
	temps	87.4739	96.2522	132.6551	140.5257	145.5216	149.7815
	saut	9.9204e-11	2.6667e-9	6.7354e-9	4.5624e-9	4.6436e-9	4.7778e-9
$p = 4$ $q = 6$	Iter	19(5)[19(5)]	22(8)	48(35)	254(225)	486(460)	2353(2327)
	temps	49.4668	57.4083	85.4365	134.0970	135.3145	142.0024
	saut	8.8477e-9	3.8165e-9	5.3221e-9	4.0890e-9	6.1453e-9	7.4250e-9
$p = 4$ $q = \ln n$	Iter	21(8)	27(8)	52(35)	246(223)	489(465)	2370(2346)
	temps	54.6208	70.0362	89.2137	118.2628	126.7116	131.3210
	saut	1.0147e-10	9.1852e-9	2.5593e-9	6.2164e-9	4.2752e-9	5.5667e-9

Table 5 : Taille $n = 2m = 50$.

par (p,q)		$\theta = 0.99$	$\theta = 0.95$	$\theta = 0.5$	$\theta = 0.1$	$\theta = 0.05$	$\theta = 0.01$
$p = e^{-n}$ $q = \ln n$	Iter	17(5)	21(8)	33(33)	218(218)	447(447)	2277(2277)
	temps	179.9027	208.4449	156.0598	156.2147	154.8138	164.0996
	saut	4.3494e-9	1.4281e-9	2.9104e-9	2.9104e-9	2.9104e-9	2.9104e-9
$p = e^{-n}$ $q = e^{-n}$	Iter	22(6)	17(8)	35(35)	224(224)	460(460)	2348(2348)
	temps	194.4674	151.0659	149.6276	155.0235	156.8934	162.1729
	saut	1.9515e-10	7.4174e-9	2.9104e-9	2.9104e-9	2.9104e-9	2.9104e-9
$p = \ln n$ $q = \ln n$	Iter	17(5)	24(8)	52(35)	245(222)	481(457)	2431(2406)
	temps	149.3048	210.0054	304.0958	418.4761	420.4186	454.8066
	saut	8.2236e-9	3.7890e-9	3.1638e-9	8.7412e-9	7.9295e-9	3.7858e-9
$p = \ln n$ $q = 6$	Iter	23(6)	24(8)	49(34)	254 (228)	481(455)	2428 (2401)
	temps	200.7371	210.6351	296.6956	456.2876	457.2768	485.2125
	saut	2.6643e-10	3.8642e-9	9.2097e-9	5.0930e-9	9.6549e-9	4.3313e-9
$p = 2$ $q = 6$	Iter	31(6)	38(8)	59(33)	248(221)	480(453)	2287(2260)
	temps	270.2373	335.7923	497.4974	485.6168	487.6024	507.5139
	saut	1.2400e-10	4.4927e-9	8.4192e-9	5.7750e-9	5.8986e-9	9.6570e-9
$p = 4$ $q = 6$	Iter	22(6)	22(8)	49(34)	254(228)	485(459)	2349(2323)
	temps	198.6114	200.4566	304.5063	469.0450	469.5445	484.3390
	saut	1.7281e-10	4.7707e-9	9.1940e-9	5.1112e-9	7.9764e-9	9.6381e-9
$p = 4$ $q = \ln n$	Iter	23(6)	32(9)	52(35)	246(223)	489(465)	2346(2322)
	temps	214.0079	289.2532	304.5222	417.0348	439.8998	439.0627
	saut	1.4829e-10	2.0180e-10	3.1447e-9	7.9163e-9	5.4498e-9	8.9497e-9

3.1.1 commentaires

À partir des tableaux précédents, les différents choix des paramètres p et q offrent une solution optimale de (P) et (D) avec un nombre d'itérations raisonnable dépend au choix

du paramètre θ (plus que θ est proche de 1 plus que le nombre des itérations décroît).

On constate aussi que la complexité algorithmique est la meilleure pour $p = e^{-n}$ et $q = \log n$.

3.2 Comparaison des algorithmes

Dans cette partie, nous proposons une comparaison entre les résultats numériques obtenus pour les différentes fonctions noyau suivantes

1. La première fonction noyau est de type logarithmique appelée fonction noyau classique, notée ψ_c , définie dans [10] par

$$\psi_c = \frac{t^2 - 1}{2} - \log t.$$

2. La deuxième fonction noyau est de type exponentielle, notée ψ_e définie dans [?] par :

$$\psi_e(t) = \frac{t^2 - 1}{2} + e^{\frac{1}{t}-1} - 1.$$

3. La troisième fonction noyau, notée ψ_t est la fonction noyau trigonométrique définie dans [?] par :

$$\psi_t(t) = \frac{t^2 - 1}{2} + \frac{4}{\pi p} \left(\tan^p \left(\frac{\pi}{2t + 2} \right) - 1 \right) \text{ pour } p = \frac{\log n}{2} - 1.$$

4. Et la dernière fonction noyau est notre nouvelle fonction noyau, notée par ψ_{new} .

Notre étude été faite pour les valeurs des paramètres p et q suivantes : $p = e^{-n}$ et $q = \log n$.

Les tableaux suivants résumant les résultats obtenus pour chaque fonction noyau avec $\theta \in \{0.1; 0.95\}$.

Table 6 : $\theta = 0.1$.

taille (m,n)	ψ_c	ψ_e	ψ_t	ψ_{new}
Iter	194(194)	192(192)	189(189)	190(190)
(5, 10) temps	0.0169	0.0184	1.2715	1.2761
saut	9.3132e-9	9.3132e-9	9.3132e-9	9.3132e-9
Iter	206(206)	408(198)	202(202)	204(204)
(10, 20) temps	0.1025	0.0881	2.9717	2.9862
saut	4.6566e-9	9.3992e-9	4.6566e-9	4.6566e-9
Iter	205(205)	1538(212)	202(202)	204(204)
(15, 30) temps	0.7879	0.5616	10.1579	10.9104
saut	6.9849e-9	9.4374e-9	6.9849e-9	6.9849e-9
Iter	205(205)	1904(215)	202(202)	205(205)
(20, 40) temps	4.3212	2.4098	42.1890	41.8870
saut	9.3132e-9	4.5846e-9	9.3132e-9	9.3132e-9
Iter	214(212)	1789(217)	336(214)	218(218)
(25, 50) temps	8.7237	8.7600	8.6321	156.2147
saut	4.9750e-9	9.4121e-9	6.8883e-9	2.9104e-9

Table 7 : $\theta = 0.95$.

taille (m,n)	ψ_c	ψ_e	ψ_t	ψ_{new}
Iter	16(7)	18(7)	25(5)	16(7)
(5, 10) temps	0.0221	0.0217	0.4644	0.3207
saut	8.6324e-9	5.9311e-9	8.2021e-10	8.5481e-9
Iter	23(8)	44(8)	20(5)	17(8)
(10, 20) temps	0.1540	0.1150	2.5571	2.3529
saut	1.1571e-9	6.6908e-10	2.5762e-9	1.1642e-9
Iter	23(8)	59(8)	21(5)	17(8)
(15, 30) temps	0.5686	0.5656	13.2353	10.5646
saut	1.7357e-9	1.8326e-9	1.0440e-9	2.1836e-9
Iter	23(8)	130(8)	24(5)	19(8)
(20, 40) temps	2.8046	2.4605	59.1370	48.0840
saut	2.3142e-9	2.4626e-9	2.6131e-9	1.9071e-9
Iter	19(8)	135(8)	20(5)	21(8)
(25, 50) temps	8.7256	8.7519	2.5571	208.4449
saut	2.8556e-9	3.0526e-9	2.5762e-9	1.4281e-9

3.2.1 commentaires

À travers les tests numériques réalisés, l'algorithme converge vers une solution optimale de (P) et (D) pour les quatres fonctions noyau.

On voit aussi que la complexité de la fonction noyau ψ_t est la meilleure pour $\theta = 0.1$ par contre pour $\theta = 0.95$ la meilleure complexité algorithmique est de notre nouvelle fonction noyau .

Conclusion

Dans ce travail, nous avons proposé une nouvelle fonction noyau efficace à deux paramètres p et q avec un terme barrière trigonométrique.

Nous avons étudié les propriétés de cette fonction noyau ainsi les effets de ses paramètres. Ainsi, analysé la complexité de l'algorithme de points intérieurs primal-dual pour les méthodes à court- et long-pas.

De plus, cette fonction nous a permis d'améliorer la complexité algorithmique de ces méthodes. Nous avons montré aussi que, le nombre d'itérations des méthodes à long-pas basées sur cette fonction noyau est $\mathcal{O}\left(pn^{\frac{(p+2)(q+1)}{2(p+1)q}} \log \frac{n}{\epsilon}\right)$. Lorsque p et q dépendent de n , on obtient la valeur minimale de la complexité algorithmique. En particulier, si $p = q = \log n$, nous obtenons la meilleure complexité à savoir $\mathcal{O}\left(\sqrt{n}(\log n) \log \frac{n}{\epsilon}\right)$. Ce résultat de complexité est le meilleur jusqu'à présent pour les méthodes de points intérieurs à long-pas basées sur les fonctions noyau de type trigonométrique.

Pour les méthodes à court-pas, nous avons obtenu la meilleure complexité, à savoir $\mathcal{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$, en prenant tout simplement $(p + q) = \mathcal{O}(1)$.

Bibliographie

- [1] **N. Anane**, Méthodes de point intérieurs pour la programmation linéaire basées sur les fonctions noyau, mémoire de magister, Université de Farhat abbas-Sétif,(2012).
- [2] **Bai YQ, El Ghami M, Roos C**, A new efficient large-update primal-dual interior point method based on finite barrier. SIAM J OPTim 13(3) :766-782 (2003).
- [3] **Bai YQ, El Ghami M, Roos C**, A comparative study of kernel function for primal-dual interior point algorithms in linear optimization. SIAM J OPTim 15(1) :101-128 (2005).
- [4] **Bouafia M, Benterki D, Yassine A**, An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a trigonometric barrier term. J OPTim Theory Appl 170 :528-545 (2016).
- [5] **Bouafia M, Yassine A**, An efficient twice parameterized trigonometric kernel function for linear optimization, Optimization and engineering (2019). <http://doi.org/10.1007/s11081-019-09467-w>.
- [6] **El Ghami M, Guennoun ZA, Bouali S, Steihaug T**, Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term. J Comput Appl Math 236 :3613-3623 (2012).

- [7] **Menniche L**, Étude théorique d'une classe de méthodes de points intérieurs pour la programmation linéaire. Thèse de do Doctorat, université de sétif (2017).
- [8] **Minoux M**, Programmation mathématique : théorie et algorithmes. Tome 1 Dunod. Paris(1983).
- [9] **Peng J, Roos C, Terlaky T**, Self -regularity : a new paradigm for primal-dual interior-point algorithms. Princeton University Press, Princeton (2002).
- [10] **Roos C, Terlaky T, Vial J. Ph**, Theory and Algorithms for linear optimization, An interior Approach, John Wiley Sons, Chichester, U. K., 1997.
- [11] **Touil I**, Étude comparative des performances d'une méthode de pointes intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire, mémoire de magister, Département de mathématiques, université de FERHAT ABBAS, Sétif (2005).
- [12] **Touil I**, Étude théorique et numérique des méthodes de points intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire. Thèse de Doctorat, université de Sétif (2017).