

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE FERHAT ABBAS SETIF-1
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES**

THESE

Présentée par :

Mousaab BOUAFIA

Pour obtenir le diplôme de doctorat en Sciences

OPTION

MATHEMATIQUES APPLIQUEES

THEME

***Etude asymptotique des méthodes de points
intérieurs pour la programmation linéaire***

**Soutenue Le : 03 mai 2016
devant le jury composé de :**

Président	: Mr. N. BENSALEM	Prof. Université Ferhat Abbas Sétif-1.
Rapporteurs	: Mr. D. BENTERKI	Prof. Université Ferhat Abbas Sétif-1.
	: Mr. A. YASSINE	Prof. Université du Havre France.
Examineurs	: Mr. J. P. CROUZEIX	Prof. Université Clermont-Fd France.
	: Mr. A. R. MAHJOUB	Prof. Université Paris Dauphine France.
	: Mr. B. BENAHMED	Prof. Ecole Nationale Polytechnique d'Oran.

Année 2015/ 2016

Je dédie cette thèse à mon défunt père : ABD ALHAMID (RACHID) qui a été le premier à m'encourager à aller si loin dans les études.

Il m'a inculqué le goût du travail, de la rigueur et de l'ambition.

Parce que tu m'avais toujours soutenu, même au plus fort de ta maladie, tu me disais et ça sera la dernière fois :

« Concentre toi à ton but, ne t'inquiète pas pour moi »

j'ai voulu le mener à terme pour que tu sois fier de moi. Merci papa, merci pour tout.

Remerciements

J'espère avoir, au cours de mes études, et particulièrement de ces dernières années, remercié les personnes qui ont compté pour moi. Cependant, ces quelques lignes me donnent l'occasion de réitérer ces remerciements pour certains et de les présenter peut-être pour la première fois à d'autres.

Pour le soutien qu'ils m'ont accordé, je remercie vivement Messieurs

- Adnan YASSINE, Professeur à l'université du HAVRE,
- Djamel BENTERKI, Professeur à l'université Sétif 1,

d'avoir accepté d'encadrer cette thèse et également de m'avoir accueilli au sein de **LMAH** (Laboratoire de Mathématiques Appliquées du Havre) où j'ai effectué une grande partie de mon travail.

J'exprime mon respect à Monsieur Naceurdine BENSLEM, Professeur à l'université Sétif 1, qui m'a fait l'honneur.

Je désire remercier également

- Jean-Pierre CROUZEIX, Professeur à l'université Blaise Pascal – Clermont Ferrand,
- Ali Ridha MAHJOUB, Professeur à l'université Paris Dauphine,
- Boubakeur BENAHMED, Professeur à l'Ecole Nationale Polytechnique d'Oran,

d'avoir accepté de juger ce travail et d'en être les examinateurs.

Je remercie l'équipe d'optimisation du laboratoire de mathématiques appliquées du Havre pour leurs aides continues et leurs encouragements, particulièrement : Docteur. Hamdi DKKIL, Mlle. Sarra TFAYLI, Mlle. Mira KHORBATLI,...

Mes remerciements s'adressent également à toute l'équipe administrative des départements de mathématiques de Sétif-1 et du Havre d'avoir mis à notre service tous les moyens disponibles, en particulier le comité et le conseil scientifique d'avoir entrepris avec une grande souplesse les démarches nécessaires pour la soutenance. Je remercie également l'université de Guelma de m'avoir octroyer des stages de courte durée qui m'ont permis d'avancer davantage dans mes travaux de thèse.

Je remercie mes chers amis sur tout l'Algérie particulièrement les amis de Hammam N'bails, aussi je remercie mes chers amis à l'étranger particulièrement l'équipe du Cedre pour leur aide précieuse durant ces années de recherche.

Je ne remercierais jamais assez certaines personnes, mais je vais essayer : je remercie infiniment mon père, ma mère, mon frères et ma sœurs pour leur soutien constant, leur encouragement, leurs sentiments et leurs prières pour moi.

Mousaab BOUAFIA

ملخص: في هذا البحث نهتم بدراسة السلوك التقاربي لخوارزميات النقطة الداخلية للبرمجة الخطية. اعتمادا على أبحاث شريجار و بادبارغ، خفضنا في عدد خطوات الخوارزمية التساقطية لكارماركار باقتراحنا خطوتين ثابتتين للانتقال. الأولى تعطي تخفيضاً في عدد خطوات الخوارزمية. أما الثانية فهي أحسن خطوة ثابتة للانتقال محصل عليه إلى يومنا هذا. كما اقترحنا مقاربتين بوسيط لخوارزمية المسار المركزي المعتمد على دالة النواة. الأولى تعمم دالة النواة التي أتى بها باي و رفقائه. أما الثانية فهي أول دالة نواة بحاجز مثلثي تعطي أحسن تكلفة للخوارزمية محصل عليه إلى يومنا هذا. طرحنا هذا مدعم بدراسات نظرية، خوارزمية وبتجارب عددية مختلفة.

كلمات مفتاحية: البرمجة الخطية، طريقة كارماركار، طرق النقط الداخلية، مقارنة يو-لوستيغ، الدالة المهيمنة، دالة النواة، تكلفة الخوارزمية.

Résumé: Dans cette recherche, on s'intéresse à l'étude asymptotique des méthodes de points intérieurs pour la programmation linéaire. En se basant sur les travaux de Schrijver et Padberg, nous proposons deux nouveaux pas de déplacement pour accélérer la convergence de l'algorithme de Karmarkar et réduire sa complexité algorithmique. Le premier pas est une amélioration modérée du comportement de l'algorithme, le deuxième représente le meilleur pas de déplacement fixe obtenu jusqu'à présent.

Ensuite nous proposons deux approches paramétrées de la l'algorithme de trajectoire centrale basé sur les fonctions noyau. La première fonction généralise la fonction noyau proposé par Y. Q. Bai et al., la deuxième est la première fonction noyau trigonométrique qui donne la meilleure complexité algorithmique, obtenue jusqu'à présent.

Ces propositions ont apporté des nouvelles contributions d'ordre algorithmique, théorique et numérique.

Mots Clés : Programmation Linéaire, Méthode de Karmarkar, Méthode de Points Intérieurs, Approche de Ye-Lustig, Fonction Potentiel, Fonction Noyau, Complexité Algorithmique.

Abstract: In this research, we are interested by asymptotic study of interior point methods for linear programming. By basing itself on the works of Schrijver and Padberg, we propose two new displacement steps to accelerate the convergence of Karmarkar's algorithm and reduce its algorithmic complexity. The first step is a moderate improvement of the behaviour of this algorithm; the second represents the best fixed displacement step obtained actually.

We propose two parameterized approaches of the central trajectory algorithm via a kernel function. The first function generalizes the kernel function given by Y. Q. Bai et al., the second is the first trigonometric kernel function that gives the best algorithmic complexity, obtained until now.

These proposals have made new contributions of algorithmic, theoretical and numerical order.

Key Words: Linear Programming, Karmarkar's Method, Interior Points Methods, Ye-Lustig Approach, Potential Function, Kernel Function, Algorithmic Complexity.

Table des matières

Introduction générale	5
1 Préliminaires	10
1.1 Eléments d'analyse convexe	10
1.1.1 Ensembles et fonctions convexes	10
1.1.2 Caractérisation d'une fonction convexe différentiable	12
1.1.3 Formules de Taylor	13
1.2 Programmation mathématique	14
1.2.1 Classification et résolution d'un programme mathématique	15
1.2.2 Existence et Unicité de solution	15
1.2.3 Conditions d'optimalité	16
1.2.4 Dualité Lagrangienne	16
1.3 Algorithme d'optimisation	18
1.3.1 Description	18
1.3.2 Convergence	18
1.3.3 Taux de convergence	18
1.4 Programmation linéaire (<i>PL</i>)	20
2 Méthode projective de type points intérieurs à court-pas et ses propriétés pour la programmation linéaire	24
2.1 Introduction	24

2.2	Algorithme de Karmarkar	26
2.3	Transformation de Karmarkar sur S_n	27
2.4	Algorithme de Ye-Lustig	30
2.5	Convergence	32
2.6	Amélioration de la convergence de l'algorithme de Karmarkar	34
2.6.1	Première amélioration du pas de déplacement	35
2.6.2	Deuxième amélioration du pas de déplacement	40
2.7	Expérimentations numériques	45
2.7.1	Exemple avec taille fixe	45
2.7.2	Exemple avec taille variable	53
2.8	Conclusion	56
3	Méthode de Trajectoire Centrale (TC) pour la programmation linéaire	57
3.1	Introduction	57
3.2	Méthode de Trajectoire Centrale classique	58
3.2.1	Présentation de la méthode	58
3.2.2	Concept de proximité	60
3.3	Méthode de Trajectoire Centrale avec poids	61
3.3.1	Présentation de la méthode	61
3.3.2	Etude de la convergence	65
3.3.3	Conclusion	65
3.4	Méthode de Trajectoire Centrale via une fonction noyau	66
3.4.1	Présentation de la méthode	66
3.4.2	Fonctions noyau et propriétés	71
3.4.3	Qualification d'une fonction noyau	73
3.4.4	Propriétés et relation entre les conditions de qualification	74
3.4.5	Borne supérieure de $\Phi(v)$ pour chaque itération externe	79
3.4.6	Analyse de la décroissance de la fonction barrière de proximité Φ	82
3.4.7	Analyse de la complexité algorithmique de l'algorithme	90

3.5	Conclusion	94
4	Famille des nouvelles fonctions noyau et algorithmes (TC) corespondant	95
4.1	Introduction	95
4.2	Fonctions noyau avec un terme barrière exponentiel	99
4.2.1	Propriétés de fonction noyau	99
4.2.2	Analyse de la complexité	105
4.2.3	Expérimentations numériques	109
4.2.4	Conclusion	110
4.3	Fonction noyau avec terme barrière trigonométrique	111
4.3.1	Propriétés de la fonction noyau	111
4.3.2	Eligibilité de la nouvelle fonction noyau	113
4.3.3	Analyse de la complexité	119
4.3.4	Expérimentations numériques et étude comparative	123
4.3.5	Conclusion	127
	Conclusion générale	128
	Annexes	129
	Annexe 1 : Calcul de la projection dans l'algorithme de Ye-Lustig	129
	Annexe 2 : Calcul d'une solution initiale primale strictement réalisable	129
	Annexe 3 : Calcul d'une solution initiale duale strictement réalisable	131
	Annexe 4 : Calcul des pas de déplacement variables dans l'algorithme de Ye-Lustig	132
	Annexe 5 : Calcul des pas de déplacement variables dans la méthode (TC)	133

Annexe 6 : Estimation de la décroissance de la fonction barrière dans la méthode de Trajectoire Centrale via une fonction noyau	134
Bibliographie	135

Introduction générale

Les premières méthodes de type "points intérieurs" et leur complexité polynomiale sont apparues au milieu des années cinquante. Ces méthodes sont générées essentiellement par une méthode barrière, proposée par Frisch [17], pour résoudre des problèmes non linéaires. En 1967, P. Huard a introduit la méthode des centres pour résoudre des problèmes possédant des contraintes non linéaires [24].

Le terme "points intérieurs" sera introduit dans les années soixante dans le fameux livre de Fiacco et McCormick [29]. Ces derniers ont développé la méthode barrière pour la programmation non linéaire convexe.

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

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

En 1979, L. G. Khachiyan a appliqué la méthode de l'ellipsoïde (initialement introduite par N. Shor en 1970 [49]) à la programmation linéaire et a prouvé que sa complexité algorithmique est de type polynomial [29]. Cette découverte représente un succès théorique, cependant en pratique l'algorithme est loin de rivaliser la méthode du simplexe.

La révolution des méthodes de points intérieurs a démarré en 1984. Karmarkar [25] découvre une méthode de points intérieurs (méthode projective) de complexité polynomiale plus efficace en pratique que celle de Khachiyan. Il annonce également des performances supérieures à celles de l'algorithme simplexe dans le cas où les matrices sont creuses et de grandes dimensions. Depuis, les méthodes de points intérieurs ont connu un développement exceptionnel donnant lieu à un changement radical dans l'étude théorique et la résolution numérique des programmes mathématiques linéaires. On parvient même à résoudre convenablement des problèmes non traités auparavant.

Ces programmes rivalisant n'échappent pas tout de même aux inconvénients théoriques, algorithmique et numérique, représentés par le problème d'initialisation qui devient de plus en plus difficile et la complexité algorithmique, devenue très importante pour l'étude de la vitesse de convergence des algorithmes développés.

En 1994, Y. Nesterov et A. Nemirovski publient une étude sur les méthodes de points intérieurs polynomiales appliquées à la programmation convexe [36]. En 2000, après seize ans à l'apparition de l'algorithme de Karmarkar, plus de 3000 articles de recherche portant sur les méthodes de points intérieurs ont été publiés par la communauté scientifique, ainsi que plusieurs ouvrages de référence (voir par exemple [45, 53]).

Les problèmes de complexité algorithmique sont au centre de notre étude dans cette thèse. Ils s'affichent depuis des années au menu de la recherche récente dans le domaine de la programmation mathématique. Pour remédier à cet inconvénient, une alternative particulièrement intéressante est celle de trajectoire centrale pour la programmation linéaire (*PL*). Il s'agit en fait de démarrer l'algorithme par un point strictement réalisable, ce qui devrait entraîner une amélioration considérable du comportement numérique.

Nos efforts, en programmation linéaire, sont concentrés sur l'aspect numérique : un algorithme de Karmarkar est soigneusement implanté avec des aménagements pour le calcul du pas de déplacement. Nous avons proposé deux nouveaux pas de déplacement pour accélérer la convergence de l'algorithme de Karmarkar et réduire sa complexité algorithmique. Le premier pas est meilleur que celui introduit par Schrijver [7, 47], et le

deuxième représente le meilleur pas de déplacement fixe obtenu jusqu'à présent [12].

Le premier pas de déplacement est défini comme suit :

$$\alpha_M = \frac{5 + nr^2}{5 + (5 + nr^2)nr}.$$

Nous avons montré que l'algorithme correspondant converge après

$$\frac{n}{1 - \log 2 + (\frac{nr^2}{10})} \log \left(\frac{c^t e_n}{\epsilon} \right) \text{ itérations.}$$

Le deuxième pas de déplacement est défini comme suit :

$$\alpha_B = \frac{n}{2n - 1}.$$

Nous avons montré que l'algorithme correspondant converge après

$$\frac{n}{1 - \log 2 + (\frac{nr^2}{2})} \log \left(\frac{c^t e_n}{\epsilon} \right) \text{ itérations.}$$

Ces nouveaux résultats théoriques sont consolidés par des bons résultats de différentes expérimentations numériques réalisées sur plusieurs exemples.

Nous sommes ensuite intéressés par la direction de déplacement dans l'algorithme de trajectoire centrale via une fonction noyau pour résoudre un problème linéaire.

Nous avons proposé deux nouvelles fonctions noyau :

La première fonction noyau à terme exponentiel généralisant celle de Y. Q. Bai et al. [3], définie comme suit :

$$\psi_M(t) = \frac{p}{2}t^2 + \exp(p(\frac{1}{t} - 1)) - (1 + \frac{p}{2}), \quad p > 0.$$

Nous avons montré que l'algorithme correspondant converge après

$$\mathbf{O} \left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon} \right) \text{ itérations pour les méthodes à long-pas ;}$$

Et

$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à court-pas.

La deuxième fonction noyau avec terme barrière trigonométrique est définie comme suit :

$$\psi_B(t) = \frac{t^2 - 1}{2} + \frac{4}{\pi p} [\tan^p h(t) - 1], \quad h(t) = \frac{\pi}{2t + 2}, \quad p \geq 2.$$

Nous avons montré que l'algorithme correspondant converge après

$\mathbf{O}\left(p n^{\frac{p+2}{2(p+1)}} \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à long-pas ;

Et

$\mathbf{O}\left(p^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à court-pas.

Nous montrons qu'en prenant la valeur $p = \left(\frac{\log n}{2} - 1\right)$, nous obtenons la meilleure complexité algorithmique, obtenue jusqu'à présent, pour les méthodes à long-pas (voir par exemple [39]), à savoir $\mathbf{O}\left(\sqrt{n} (\log n) \log \frac{n}{\epsilon}\right)$. En plus, la fonction ψ_B , que nous proposons, est la première fonction noyau trigonométrique qui donne cette meilleure borne (voir [19, 3, 42, 41, 30, 33, 11]).

Les résultats tests effectués sur plusieurs exemples confirment les propos théoriques démontrés.

Les résultats obtenus sont encourageants, ce qui nous a motivé à généraliser nos résultats à plusieurs problèmes d'optimisation tels que le problème semi-défini (*SDP*), les problèmes de complémentarité linéaire et les problèmes convexes avec des contraintes linéaires.

La thèse est décomposée de quatre chapitres. Dans le premier, on présente des notions de base sur l'analyse convexe et la programmation mathématique et les résultats d'existence d'une solution optimale qui serviront d'appuis pour la suite. Le second chapitre est basé sur les travaux de Karmarkar [25], Schrijver [47] et Padberg [37]. Dans ce chapitre, nous proposons des nouveaux pas de déplacement permettant d'améliorer le

comportement numérique de l'algorithme de Karmarkar. Notre étude est d'ordre théorique, algorithmique et numérique. Le troisième chapitre représente une synthèse théorique et une étude algorithmique pour la résolution d'un programme linéaire primal-dual par les méthodes de points intérieurs basées sur la méthode de trajectoire centrale. Nous avons développé des nouvelles méthodes de trajectoire centrale en injectant des fonctions noyau.

Enfin, dans le quatrième chapitre, nous proposons deux nouvelles fonctions noyau ayant des caractéristiques spécifiques. La première fonction nous a permis de généraliser les résultats obtenus par Y. Q. Bai et al. dans [3], tandis que la deuxième nous a permis d'améliorer tous les résultats théoriques obtenus jusqu'à présent et de fournir la meilleure borne de la complexité algorithmique des méthodes à long-pas. A la fin, nous présentons des annexes qui montrent le calcul des paramètres d'amélioration de l'efficacité et la performance des algorithmes développés pour la résolution de la programmation linéaire.

Chapitre 1

Préliminaires

L'objectif de ce premier chapitre est de présenter quelques notions fondamentales de l'analyse convexe, de la programmation linéaire et l'étude asymptotique de la convergence d'un algorithme d'optimisation. Ces notions sont utiles pour démontrer les résultats théoriques dans les chapitres suivants.

1.1 Eléments d'analyse convexe

Dans cette section, on présente un rappel des notions fondamentales de l'analyse convexe qui serviront d'appuis pour la suite. Pour plus de détails voir [27, 44].

1.1.1 Ensembles et fonctions convexes

- Un 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 :

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

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 :

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

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

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

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

• 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 convexe sur C 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 :

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

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

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

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

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

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

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

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

1.1.3 Formules de Taylor

Soit $\Omega \subset \mathbb{R}^n$ ouvert, $f : \Omega \longrightarrow \mathbb{R}$, $a \in \Omega$ et $h \in \mathbb{R}^n$ tels que $[a, a + h] \subset \Omega$. 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

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

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

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

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 $\theta \in [0, 1]$ telque $f(a+h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a+\theta h) h, h \rangle$.

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

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

Remarque 1.1.1 Dans les formules précédentes, la notation $\mathbf{o}(\|h\|^k)$ pour $k \in \mathbb{N}^*$ signifie une expression qui tend vers 0 plus vite que $\|h\|^k$ (c'est à dire, si on la divise par $\|h\|^k$, le résultat tend vers 0 quand h tend vers 0).

1.2 Programmation mathématique

La programmation mathématique constitue un domaine vaste et riche dans l'analyse numérique. Elle traite plusieurs modèles mathématiques et problèmes pratiques importants.

1.2.1 Classification et résolution d'un programme mathématique

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

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

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 ou ensemble des contraintes ou tout simplement "le domaine".

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).
- La programmation convexe (f, g_i convexes, h_j affines, C convexe).
- La programmation en nombres entiers (C est un ensemble discret, c'est à dire les variables sont entières).

1.2.2 Existence et Unicité de solution

Dans ce paragraphe, nous donnons les deux théorèmes d'existence et le théorème d'unicité les plus utilisés (voir [8, 27]) :

Théorème 1.2.1 *Si C est compact non vide de \mathbb{R}^n et si f est continue sur C alors (PM) admet au moins une solution optimale $x^* \in C$.*

Théorème 1.2.2 *Si C est fermé non vide de \mathbb{R}^n , f est continue et coercive sur C (c'est-à-dire $\lim f(x) = +\infty$ lorsque $\|x\|$ tend vers $+\infty$) alors (PM) admet au moins une solution optimale.*

Théorème 1.2.3 *Si C est convexe non vide de \mathbb{R}^n , f est strictement convexe sur C alors (PM) admet une solution optimale au plus.*

1.2.3 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 $g_i(\bar{x}) = 0$.

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

1.2.4 Dualité Lagrangienne

Soit :

$$S = \{x \in D \subset \mathbb{R}^n : g_i(x) \leq 0, \quad i = 1, \dots, k, \quad h_j(x) = 0, \quad j = 1, \dots, m\}$$

et on considère le problème primal :

$$m = \inf_x [f(x), \quad x \in S]$$

Le Lagrangien associé à ce problème est la fonction $L : D \times [0, +\infty[^k \times \mathbb{R}^m \longrightarrow \mathbb{R}$, définie par :

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^k \lambda_i g_i(x) + \sum_{j=1}^m \mu_j h_j(x)$$

On pose :

$$\alpha(x) = \sup_{\lambda, \mu} [L(x, \lambda, \mu) : \lambda \geq 0] = \begin{cases} f(x) & \text{si } g_i(x) \leq 0 \text{ et } h_j(x) = 0 \\ +\infty & \text{sinon} \end{cases}$$

donc :

$$\bar{\alpha} = \inf_{x \in D} \alpha(x) = \inf_x [f(x) : x \in S]$$

Le problème dual associé au problème primal est :

$$\bar{\beta} = \sup_{(\lambda, \mu)} \inf_{x \in D} [L(x, \lambda, \mu)]$$

On a l'inégalité de dualité

$$-\infty \leq \bar{\beta} \leq \bar{\alpha}$$

Théorème 1.2.4 (*Karush - Kuhn - Tucker*)[26] :

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 pour (PM), alors il existe des réels dits multiplicateurs de Lagrange :

$\mu_i \in \mathbb{R}^+, i = 1, \dots, m$ et $\lambda_j \in \mathbb{R}, j = 1, \dots, p$ tels que :

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

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).

1.3 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 donnerons ici quelques définitions.

1.3.1 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 :

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

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.

1.3.2 Convergence

Définition 1 :

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^* .

1.3.3 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 ($e_k = \|x_k - x^*\|$).

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

Définition 2 (Notation **O**) Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$. On note $f(n) = \mathbf{O}(g(n))$ 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 3 (Notations **o**, **Ω** , **Θ**) 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 $f(n) = \mathbf{\Omega}(g(n))$ lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

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

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

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 $\|e_k\| = \mathbf{\Omega}(\|e_{k+1}\|)$ 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 :

$$\exists c \in [0, 1[, \exists k_0 \geq k, e_{k+1} \leq ce_k$$

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

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

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

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

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

1.4 Programmation linéaire (PL)

La programmation linéaire dans \mathbb{R}^n est constituée de la famille de programmes qui traitent la résolution des problèmes d'optimisation pour lesquels la fonction objectif et toutes les contraintes sont linéaires. Il s'agit des problèmes d'optimisation les plus célèbres, les plus simples théoriquement et les plus étudiés en recherche opérationnelle.

Le problème suivant est appelé programme linéaire :

$$(PL) \left\{ \begin{array}{l} \min \quad c^t x \\ \quad \quad s.c \\ Ax = b \\ Dx \geq e \\ x \in \mathbb{R}^n. \end{array} \right.$$

où $A \in \mathbb{R}^{m \times n}$ et $D \in \mathbb{R}^{p \times n}$ sont des matrices données, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et $e \in \mathbb{R}^p$ sont des vecteurs donnés.

On peut montrer que tout programme linéaire peut se ramener à l'une des deux formes suivantes :

a)Forme canonique :

$$(PLC) \left\{ \begin{array}{l} \min \ c^t x \\ s.c \\ Ax \geq b \text{ (ou } \leq) \\ x \geq 0. \end{array} \right.$$

b)Forme standard :

$$(PLS) \left\{ \begin{array}{l} \min \ c^t x \\ s.c \\ Ax = b \\ x \geq 0. \end{array} \right.$$

où A est une matrice réelle de type (m, n) supposée de plein rang (c'est-à-dire : $\text{rg}(A) = m < n$), b un vecteur de \mathbb{R}^m .

Dans toute la suite, on s'intéresse au problème (PL) sous forme standard suivant :

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

Le dual du programme linéaire (PL) est un programme linéaire défini par :

$$(DL) \left\{ \begin{array}{l} \max \ b^t y \\ s.c \\ A^t y + s = c \\ s \geq 0, \ s \in \mathbb{R}^n \\ y \in \mathbb{R}^m. \end{array} \right.$$

On note par :

- $F_{(PL)} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, l'ensemble des solutions primales réalisables de (PL) .
- Un vecteur $x \in F_{(PL)}$ est appelé solution réalisable de (PL) .
- Un vecteur $x^* \in F_{(PL)}$ minimisant la fonction objectif de (PL) s'appelle solution optimale de (PL) .
- Un programme linéaire (PL) réalisable est borné si la fonction objectif est bornée sur $F_{(PL)}$.
- $\overset{\circ}{F}_{(PL)} = \{x \in \mathbb{R}^n : Ax = b, x > 0\}$, l'ensemble des solutions primales strictement réalisables de (PL) .
- $F_{(DL)} = \{y \in \mathbb{R}^m : A^t y + s = c, s \geq 0\}$, l'ensemble des solutions duales réalisables de (DL) .
- Un vecteur $y^* \in F_{(DL)}$ maximisant la fonction objectif de (DL) s'appelle solution optimale de (DL) .
- $\overset{\circ}{F}_{(DL)} = \{y \in \mathbb{R}^m : A^t y + s = c, s > 0\}$, l'ensemble des solutions duales strictement réalisables de (DL) .
- $\overset{\circ}{F} = \overset{\circ}{F}_{(PL)} \times \overset{\circ}{F}_{(DL)}$, l'ensemble des solutions primales-duales strictement réalisables de (PL) et (DL) .

Donnons quelques résultats fondamentaux de dualité en programmation linéaire :

- Si l'un des problèmes (PL) et (DL) admet une solution optimale, il en est de même pour l'autre, et leurs valeurs optimales correspondantes sont égales.
- Si l'un des problèmes a une valeur optimale non finie, l'autre n'a pas de solution optimale.

Théorème 1.4.1 *(Dualité faible)* Si x et (y, s) sont respectivement des solutions réalisables pour (PL) et (DL) alors,

$$c^t x \geq b^t y.$$

Théorème 1.4.2 *(Dualité forte)* Si \bar{x} et (\bar{y}, \bar{s}) sont respectivement des solutions réali-

sables correspondant une valeur optimale finie pour (PL) et (DL) telles que

$$c^t \bar{x} = b^t \bar{y},$$

alors \bar{x} est une solution primale optimale de (PL) et \bar{y} est une solution duale optimale de (DL) .

Remarque 1.4.1 On peut remarquer facilement que si \bar{x} et (\bar{y}, \bar{s}) sont respectivement des solutions réalisables de (PL) et (DL) , alors on a la propriété suivante :

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

Chapitre 2

Méthode projective de type points intérieurs à court-pas et ses propriétés pour la programmation linéaire

2.1 Introduction

Avant 1984, tout problème d'optimisation linéaire se résolvait par la méthode du simplexe, développée par Dantzig [15], ou par une variante de celle ci. Plusieurs recherches ont été menées pour mettre au point une autre méthode polynomiale plus efficace mais sans résultats. Aussi, pendant une quarantaine d'années, cette méthode domina l'optimisation linéaire. Dans les années 70, la théorie de la complexité devient une partie importante en optimisation en général, et en particulier, en optimisation linéaire. On exige aux méthodes développées de montrer que leur convergence se réalise en un temps polynomial. C'est à dire de résoudre le problème en un nombre d'opérations borné par un polynôme de la même taille du problème. Malheureusement, la méthode du simplexe n'a pas cette propriété, comme l'ont montré Klee et Minty [31]. On se demanda alors si

un autre algorithme d'optimisation linéaire avait cette propriété.

L'algorithme de Karmarkar pour résoudre les problèmes linéaire [25] est le premier algorithme réellement efficace qui résout ces problèmes en un temps polynomial. En utilisant une fonction potentiel, Karmarkar montre que son algorithme converge après un nombre $\mathbf{O}(\mathbf{nq} + \mathbf{n} \log \mathbf{n})$ itérations pour un pas de déplacement $\alpha = \frac{1}{4}$, où n est le nombre de variables et 2^{-q} est le paramètre de précision. Depuis, plusieurs chercheurs s'intéressaient à cet algorithme dans le but d'améliorer au mieux son comportement numérique. Les deux éléments de base visés sont la direction de déplacement qui domine le coût du calcul à chaque itération et le pas de déplacement qui joue un rôle important dans la vitesse de convergence. En effet, en modifiant la forme analytique de la fonction potentiel, Padberg [37] a pu réduire le nombre des itérations à $\mathbf{O}(\mathbf{nq})$ itérations pour $\alpha = \frac{1}{2}$. De son côté Schrijver [47], en gardant la même fonction potentiel de Karmarkar, a pu montré que l'algorithme converge après $\frac{n}{1-\log(2)} \log\left(\frac{c^t e_n}{\epsilon}\right)$ itérations pour $\alpha_S = \frac{1}{1+nr}$, où c est le vecteur des coûts, $e_n = (1, 1, \dots, 1)^t \in \mathbb{R}^n$, et ϵ le paramètre de précision. Dans ce travail, en utilisant la même fonction potentiel de Karmarkar, nous proposons deux nouveaux pas de déplacement : le premier pas est meilleur que celui introduit par Schrijver [47, 7], et le deuxième représente le meilleur pas de déplacement fixe obtenu jusqu'à présent [12]. Nous montrons que notre algorithme converge après $\frac{n}{1-\log(2)+(\frac{nr^2}{10})} \log\left(\frac{c^t e_n}{\epsilon}\right)$ itérations pour le premier pas de déplacement $\alpha_M = \frac{5+nr^2}{5+(5+nr^2)nr}$ et après $\frac{n}{1-\log(2)+(\frac{nr^2}{2})} \log\left(\frac{c^t e_n}{\epsilon}\right)$ itérations pour le deuxième pas de déplacement $\alpha_B = \frac{n}{2n-1}$.

Les deux familles des méthodes de points intérieurs, les plus connues en optimisation, sont les suivantes :

- La réduction du potentiel projective (algorithme de Karmarkar).
- La trajectoire centrale (**TC**).

Dans ce chapitre, on s'intéresse à une méthode de points intérieurs de type projectif, où on donne une description générale de la méthode classique de Karmarkar [25] et la variante la plus utilisée traduite par Ye-Lustig [32] ainsi que les résultats de convergence polynomiale. A la fin de ce chapitre, nous présentons nos deux nouveaux pas de

déplacement pour améliorer le taux de convergence de l'algorithme de Karmarkar. Nous présentons également des études théoriques pour montrer ces améliorations et des simulations numériques qui confirment et consolident les résultats théoriques.

2.2 Algorithme de Karmarkar

Dans son article [25], Karmarkar traite le problème linéaire sous la forme réduite suivante :

$$(PK) \begin{cases} \min c^t x = 0 \\ Ax = 0 \\ x \in S_n = \{x \in \mathbb{R}_+^n, e_n^t x = 1\}, \end{cases}$$

où $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ de plein rang ($\text{rang}(A) = m < n$) et $e_n = (1, 1, \dots, 1)^t \in \mathbb{R}^n$.

Le centre du simplexe S_n est une solution strictement réalisable de $(PK) : x^0 = \frac{e_n}{n} \in S_n \cap \ker(A)$.

Remarque 2.2.1 *On peut ramener facilement un programme linéaire général quelconque mis sous la forme standard suivante :*

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

sous la forme réduite (PK) de Karmarkar.

Pour ce faire Karmarkar [25] utilise la transformation projective suivante :

$$T_a : \mathbb{R}_+^n \longrightarrow S_{n+1}$$

$$x \mapsto T_a(x) = y$$

Telle que :

$$T_a(x) = \begin{cases} (y[n])_i = y_i = \frac{\frac{x_i}{a_i}}{1 + \sum_{i=1}^n \frac{x_i}{a_i}}, & i = 1, \dots, n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{cases}$$

Le problème (PL) se transforme alors via la transformation T_a au problème linéaire $(PL)_a$ suivant :

$$(PL)_a \begin{cases} \min (c')^t y = 0 \\ A'x = 0 \\ y \in S_{n+1} = \{y \in \mathbb{R}_+^{n+1}, e_{n+1}^t y = 1\} \end{cases}$$

Où :

- $a = (a_1, a_2, \dots, a_n)^t$ est une solution strictement réalisable de (PL) .
- $D_a = \text{diag}(a)$ est la matrice diagonale dont les éléments diagonaux sont les composantes du vecteur a .

- $A' = [AD_a, -b] \in \mathbb{R}^{m \times (n+1)}$.
- $c' = [(D_a c)^t, -z^*]^t \in \mathbb{R}^{(n+1)}$.

Remarque 2.2.2 La transformation T_a est inversible et on a :

$$T_a^{-1}(y) = x = \frac{D_a y[n]}{y_{n+1}}$$

2.3 Transformation de Karmarkar sur S_n

Cette transformation est définie à chaque itération par :

$$T_k : S_n \longrightarrow S_n \\ x \mapsto y = T_k(x) = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x}$$

tel que : $D_k = \text{diag}(x^k)$

De même, la transformation T_k est inversible et on a :

$$x = T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}$$

Le problème transformé de (PK) par la transformation T_k est le problème linéaire suivant :

$$(PKT) \left\{ \begin{array}{l} \min (D_k c)^t y \\ AD_k y = 0 \\ y \in S_n = \left\{ y \in \mathbb{R}_+^n : \sum_{i=1}^n y_i = 1 \right\}, \end{array} \right.$$

qui peut s'écrire aussi sous la forme :

$$(PKT) \left\{ \begin{array}{l} \min (D_k c)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y \geq 0 \end{array} \right.$$

Le problème transformé (PKT) est aussi mis sous la forme réduite de Karmarkar.

De même, cette transformation permet de ramener à chaque itération, l'itéré x^k au centre du simplexe S_n , i.e, $T_k(x^k) = \frac{e_n}{n}$.

Remarque 2.3.1 *Les deux transformations utilisées par Karmarkar ramènent les itérés au centre du simplexe. D'où deux stratégies sont envisagées :*

L'une est de ramener le problème général (PL) à la forme réduite (PKT) puis revenir à chaque itération au problème initial (PL) pour tester l'optimalité des itérés.

La deuxième, on suggère de travailler dans S_n jusqu'à l'optimalité (sans revenir à la forme initiale).

La première stratégie est la plus commode du point de vue numérique, traduite par la variante de Ye-Lustig dans le cas général (cas où z^ est inconnu).*

Avant d'appliquer les conditions d'optimalité, Karmarkar [25] relaxe le problème

(*PKT*) en remplaçant la condition $y \geq 0$ par la sphère $s\left(\frac{e_n}{n}, \alpha r\right)$; (on montre facilement que si $y \in s\left(\frac{e_n}{n}, \alpha r\right)$ alors $y \geq 0$, voir [28]) où $r = \frac{1}{\sqrt{n(n-1)}}$ et $0 < \alpha < 1$.

Le problème (*PKT*) devient alors :

$$(\text{PKT})_r \begin{cases} \min (D_k c)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \|y - \frac{e_n}{n}\|^2 \leq (\alpha r)^2 \end{cases}$$

En utilisant les conditions d'optimalité, la solution optimale du problème $(\text{PKT})_r$ est donnée par :

$$y = \frac{e_n}{n} - \alpha r d_k$$

où $d_k = \frac{P_k}{\|P_k\|}$; avec P_k est la projection du vecteur $D_k c$ sur la noyau de la matrice des contraintes $\begin{bmatrix} AD_k \\ e_n^t \end{bmatrix}$.

En revenant par la transformation inverse T_k^{-1} on obtient une solution réalisable x^k pour le problème initial (*PK*) tel que $x^k = T_k^{-1}(y^k) = \frac{D_k y^k}{e_n^t D_k y^k}$.

Remarque 2.3.2 La condition $z^* = 0$ (ou z^* est connu) posée par Karmarkar est très restrictive en pratique, pour cela plusieurs variantes sont proposées par les chercheurs qui considèrent des approximations de z^* par des bornes supérieures, inférieures ou supérieures et inférieures à la fois.

La variante la plus utilisée est celle de Ye-Lustig qui propose une approximation majorant de z^* en prenant à chaque itération $z_k = c^t x^k > z^*$. Pour plus de détails voir [28].

Dans ce qui suit, on donne l'algorithme de Ye-Lustig appliqué pour z^* inconnu.

2.4 Algorithme de Ye-Lustig

On considère dans ce paragraphe le problème linéaire générale :

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

D'après la remarque **2.3.2**, et la transformation T_a on peut ramener (PL) à la forme :

$$(PKT)_r \quad \begin{cases} \min & \left(\begin{array}{c} D_k c \\ -c^t x^k \end{array} \right)^t y \\ B_k y = 0 \\ e_{n+1}^t y = 1 \\ \left\| y - \frac{e_{n+1}}{n+1} \right\|^2 \leq (\alpha r)^2 \end{cases}$$

où $B_k = [AD_k, -b] \in \mathbb{R}^{m \times (n+1)}$.

L'algorithme correspondant de Ye-Lustig est le suivant :

Algorithme 1. Algorithme de Ye-Lustig

Début algorithme

Initialisation

$k = 0, x^0 > 0$ strictement réalisable pour (PL) (voir annexe 2, pages 130-131);

Un paramètre de précision $\epsilon > 0$;

$\|P_0\| = (1 + \epsilon) |c^t x^0|$;

Tant que $\frac{\|P_k\|}{|c^t x^0|} > \epsilon$ **faire** :

Etape 0 : (Construire la matrice des contraintes)

$B_k = [A_k, -b]$ où $A_k = AD_k$ et $D_k = \text{diag}(x^k)$;

Etape 1 : (Calculer la projection P_k)

$$P_k = \left[I - B_k^t (B_k B_k^t)^{-1} B_k \right] \begin{pmatrix} D_k c \\ -c^t x^k \end{pmatrix} \quad (\text{voir annexe 1, page 129});$$

Et poser $d_k = \frac{P_k}{\|P_k\|}$;

Etape 2 : Calculer l'itéré suivant :

$$y^k = \frac{e_{n+1}}{n+1} - \alpha r d_k, \quad r = \frac{1}{\sqrt{n(n+1)}} \quad \text{avec} \quad 0 < \alpha < 1;$$

α est le pas de déplacement (voir les sections 2.5 et 2.6);

Etape 3 : (Revenir au problème initial (PL) par T_k^{-1})

$$x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k[n]}{y_{n+1}^k}, \quad k = k + 1;$$

Fin tant que;

Fin algorithme.

Remarque 2.4.1 *L'algorithme de Ye-Lustig est de type à deux phases :*

La phase 1 (phase d'initialisation) : consiste à trouver une solution initiale strictement réalisable de (PL) .

La phase 2 : consiste à améliorer les itérés jusqu'à l'optimalité des itérés.

Remarque 2.4.2 *Dans un premier temps, Karmarkar a pris le pas de déplacement α fixe, situé entre 0 et 1. Ceci assure que tous les itérés restent à l'intérieur du simplexe et permet de montrer la convergence polynomiale de l'algorithme.*

Sur le plan numérique, il s'avère que plus α est grand ($\alpha > 1$) plus l'algorithme converge vite. Pour cette raison, plusieurs chercheurs suggèrent d'utiliser des pas variables moyennant les méthodes de recherche linéaire. Malheureusement les procédures sont coûteuses et on ne dispose pas des résultats de convergence.

D'autre part, on pense plutôt à améliorer les résultats théoriques portés par Karmarkar sur α , en utilisant de petits pas qui dépendent de la taille du problème considéré (*short step*) (travaux de Schrijver [47], R.Naseri, A.Valinejad [35]).

Pour le moment, on donne le théorème de la convergence polynomiale de l'algorithme de Karmarkar.

2.5 Convergence

Pour étudier la convergence polynomiale de l'algorithme, Karmarkar a utilisé (dans le cas $z^* = 0$) la fonction potentiel suivante :

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

définie sur :

$$D_x = \{x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1\}$$

Pour montrer la convergence de son algorithme, Karmarkar a utilisé le lemme suivant :

Lemme 2.5.1 [28]

Soit y^k une solution strictement réalisable pour (PKT) alors on a :

$$\frac{c^t D_k y^k}{c^t D_k \frac{e_n}{n}} \leq \left(1 - \alpha \frac{1}{n-1}\right) = \left(1 - \alpha \frac{r}{R}\right) = (1 - \alpha n r^2) \text{ tel que } R = \sqrt{\frac{n-1}{n}}, \quad 0 < \alpha < 1$$

Lemme 2.5.2 [25] Pour $\beta = \alpha n r < 1$ et y^k défini par l'algorithme de Karmarkar, nous avons :

$$\sum_{i=1}^n \log y_i^k \geq -\frac{\beta^2}{2(1-\beta)} - n \log n$$

Théorème 2.5.1 (*Théorème de convergence de Karmarkar*) [25]

Si $0 < \alpha_K \leq \frac{1}{4}$, alors en partant de $x^0 = \frac{e_n}{n}$, après $\mathbf{O}(\mathbf{nq} + \mathbf{n} \log \mathbf{n})$ itérations, l'algorithme trouve un point réalisable x tel que :

$$(1) \ c^t x = 0$$

ou

$$(2) \ \frac{c^t x}{c^t x^0} \leq 2^{-q} \text{ où } q \text{ est une précision fixée.}$$

Dans ces travaux, Padberg [28] a pu améliorer la convergence de l'algorithme classique de Karmarkar en modifiant la fonction potentiel de Karmarkar ainsi :

$$h(x) = \frac{c^t x}{\left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}}$$

définie sur :

$$D_x = \{x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1\}$$

Pour montrer la convergence de son algorithme, Padberg a utilisé le lemme suivant :

Lemme 2.5.3 [37] Pour $0 < \alpha < 1$ et y^k défini par l'algorithme de Karmarkar, nous avons :

$$\frac{1}{n \left(\prod_{i=1}^n y_i^k \right)^{\frac{1}{n}}} \leq \frac{1}{1 + \frac{\alpha}{n-1}} \left(\frac{1 + \frac{\alpha}{n-1}}{1 - \alpha} \right)^{\frac{1}{n}}, \quad n \geq 3$$

et a montré que pour $\alpha_P = \frac{1}{2}$, l'algorithme converge après $\mathbf{O}(\mathbf{nq})$ itérations.

Dans le même sens, et indépendamment de Padberg, Schrijver [47] a montré le théorème suivant :

Théorème 2.5.2 [47] Pour $\alpha_S = \frac{1}{1+nr}$; (n est la taille du problème (PL), et $r = \frac{1}{\sqrt{n(n-1)}}$) l'algorithme converge après $\frac{n}{1-\log(2)} \log \left(\frac{c^t e_n}{\epsilon} \right)$ itérations.

Dans la section suivante, nous proposons deux nouvelles valeurs du pas de déplacement α , meilleures que celle de Schrijver, dans le but d'améliorer davantage la vitesse de convergence de l'algorithme de Karmarkar.

2.6 Amélioration de la convergence de l'algorithme de Karmarkar

Cette section représente la partie la plus importante de notre chapitre. On s'intéresse essentiellement au développement du pas de déplacement. Notre contribution est d'ordre théorique et numérique. Du point de vue théorique, en se basant sur les travaux de Schrijver [47] et Padberg [37], nous proposons deux nouveaux pas de déplacement α_M , α_B , meilleurs que celui introduit par Schrijver.

Sur le plan numérique, nous avons effectué des tests numériques sur plusieurs exemples qui confirment les propos théoriques. Ces résultats sont présentés à la fin de ce chapitre.

Avant cela, nous avons besoin des lemmes ci-dessous, utilisés par Schrijver [46], pour faciliter au lecteur la compréhension des propos établis le long des différentes preuves.

Lemme 2.6.1 [46] *Soit ψ_n une fonction définie par :*

$$\begin{aligned} \psi_n :]-1, +\infty[^n &\rightarrow \mathbb{R}_+ \\ x &\mapsto \psi_n(x) = e_n^t x - \sum_{i=1}^n \log(1 + x_i) \end{aligned}$$

Alors pour $\|x\| < 1$ on a : $\psi_1(-\|x\|) \geq \psi_n(-x)$.

Rappelons que la fonction potentiel de Karmarkar est définie par :

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

sur l'ensemble :

$$D_x = \{x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1\}.$$

Dans les deux lemmes suivants, on donne la relation entre la fonction potentiel et la fonction objectif du problème (PK), et on montre que plus f est décroissante plus l'objectif $c^t x$ se réduit à zéro.

Lemme 2.6.2 [46] *Si $x \in D_x$, alors :*

$$c^t x \leq \exp \left(\frac{f(x)}{n} \right)$$

Lemme 2.6.3 [46] *A chaque itération k , la fonction potentiel diminue de la quantité suivante :*

$$f(x^{k+1}) - f(x^k) = -\Delta,$$

tel que :

$$x^{k+1} = T_k^{-1}(y^k) \text{ et } \Delta = n \log \frac{c^t D_k e_n}{c^t D_k y^k} + \sum_{i=1}^n \log y_i^k.$$

2.6.1 Première amélioration du pas de déplacement

Dans cette partie, en se basant sur les lemmes précédents, on donne dans le lemme suivant un nouveau pas de déplacement α_M , avec $\alpha_S < \alpha_M < \frac{1}{2}$, qui permet de réduire le nombre d'itérations nécessaires pour la convergence de l'algorithme de Karmarkar.

Lemme 2.6.4 ([7])

Si $\alpha_M = \frac{5+nr^2}{5+(5+nr^2)nr}$, l'algorithme de Karmarkar converge après $\frac{n}{1-\log(2)+(\frac{nr^2}{10})} \log \left(\frac{c^t e_n}{\epsilon} \right)$ itérations.

Remarque 2.6.1 :

- *On peut voir facilement que : $\alpha_S < \alpha_M < \frac{1}{2}$, du fait que*

$$\begin{aligned} & 5 + nr^2 > 5 \\ \Leftrightarrow & (5 + nr^2)(1 + nr) > 5 + (5 + nr^2)nr \\ \Leftrightarrow & \frac{(5+nr^2)}{5+(5+nr^2)nr} > \frac{1}{1+nr} \end{aligned}$$

ce qui donne :

$$\alpha_S < \alpha_M$$

et comme $n \geq 2$ on a :

$$\begin{aligned}
\alpha_M &= \frac{5+nr^2}{5+(5+nr^2)nr} \\
&= \frac{5(n-1)+1}{5(n-1)+[5(n-1)+1]nr} \\
&< \frac{5(n-1)+1}{5(n-1)+[5(n-1)+1]\frac{4}{3}} \\
&= \frac{15n-12}{35n-31} < \frac{1}{2}
\end{aligned}$$

ce qui donne :

$$\alpha_M < \frac{1}{2}$$

- On montre facilement que ψ_1 est croissante sur $[0, +\infty[$ et on a en particulier :

$$\psi_1\left(\frac{1}{5}nr(5+nr^2)\right) \geq \psi_1(nr) \geq \psi_1(1) = 1 - \log 2$$

- Pour prouver le lemme **2.6.4**, on a besoin du lemme suivant :

Lemme 2.6.5 ([7]) Pour $\alpha = \alpha_M = \frac{5+nr^2}{5+(5+nr^2)nr}$, on a :

a) $\psi_1(-\alpha_M nr^2) > \frac{1}{5}\alpha_M n^2 r^4.$

b) $\Delta \geq \psi_1\left(\frac{1}{5}nr(5+nr^2)\right).$

Preuve. **a)** Par définition on a :

$$\psi_1(-\alpha_M nr^2) = -\alpha_M nr^2 - \log(1 - \alpha_M nr^2).$$

Comme $\log(1-x) < -x - \frac{x^2}{2}$ pour $0 < x < 1$ alors :

$$\begin{aligned}
\psi_1(-\alpha_M nr^2) &> -\alpha_M nr^2 + \alpha_M nr^2 + \frac{\alpha_M^2 n^2 r^4}{2} \\
&> \frac{\alpha_M n^2 r^4}{2} \alpha_S > \frac{\alpha_M n^2 r^4}{2} \frac{2}{5} = \frac{1}{5}\alpha_M n^2 r^4,
\end{aligned}$$

car :

$$\alpha_S = \frac{1}{1+nr} \geq \frac{1}{1+\sqrt{2}} > \frac{1}{\left(\frac{5}{2}\right)} = \frac{2}{5}.$$

Donc :

$$\psi_1(-\alpha_M n r^2) > \frac{1}{5} \alpha_M n^2 r^4.$$

b) D'après le lemme **2.6.3** en prenant $\frac{r}{R} = n r^2$ on a :

$$\Delta \geq \alpha n^2 r^2 + n \psi_1(-\alpha n r^2) - \psi_1(-\alpha n r), \quad 0 < \alpha < \frac{1}{n r}$$

pour $\alpha = \alpha_M = \frac{5+n r^2}{5+(5+n r^2)n r}$ et d'après a) on a :

$$\begin{aligned} \Delta &> \alpha_M n^2 r^2 + n \left(\frac{1}{5} \alpha_M n^2 r^4 \right) - \psi_1(-\alpha_M n r) \\ &> \alpha_M n^2 r^2 + \frac{1}{5} n r^2 (\alpha_M n^2 r^2) + \alpha_M n r + \log(1 - \alpha_M n r) \\ &> (n^2 r^2 + \frac{1}{5} n r^2 (n^2 r^2) + n r) \alpha_M + \log(1 - \alpha_M n r) \\ &> \frac{1}{5} (5 n^2 r^2 + n r^2 (n^2 r^2) + 5 n r) \frac{5+n r^2}{5+(5+n r^2)n r} + \log \left(1 - \frac{5+n r^2}{5+(5+n r^2)n r} n r \right) \\ &> \frac{1}{5} n r (5 n r + n r^2 (n r) + 5) \frac{5+n r^2}{5+(5+n r^2)n r} + \log \left(1 - \frac{5+n r^2}{5+(5+n r^2)n r} n r \right) \\ &> \frac{1}{5} n r ((5 + n r^2) (n r) + 5) \frac{5+n r^2}{5+(5+n r^2)n r} + \log \left(\frac{5}{5+(5+n r^2)n r} \right) \\ &> \frac{1}{5} n r (5 + n r^2) + \log \left(\frac{5}{5+(5+n r^2)n r} \right) \\ &> \frac{1}{5} n r (5 + n r^2) + \log \left(\frac{1}{1+(5+n r^2)\frac{n r}{5}} \right) \\ &> \frac{1}{5} n r (5 + n r^2) - \log(1 + \frac{1}{5} n r (5 + n r^2)) = \psi_1 \left(\frac{1}{5} n r (5 + n r^2) \right) \end{aligned}$$

donc :

$$\Delta > \psi_1 \left(\frac{1}{5} n r (5 + n r^2) \right).$$

Preuve du lemme 2.6.4 :

D'après le lemme **2.6.5 b)** on a :

$$\Delta > \psi_1 \left(\frac{1}{5} n r (5 + n r^2) \right) = \psi_1 \left(n r + \frac{n r}{5} n r^2 \right) > \psi_1 \left(1 + \frac{1}{5} n r^2 \right).$$

Car $n r > 1$ et ψ_1 est croissante (remarque **2.6.1**), d'où

$$\Delta > \psi_1 \left(1 + \frac{1}{5} n r^2 \right), \tag{1.1}$$

d'autre part

$$\psi_1(1 + \frac{1}{5}nr^2) - \psi_1(1) > \frac{1}{2} \left(\frac{1}{5}nr^2 \right) = \frac{nr^2}{10} \text{ car } \log(1+x) \leq x,$$

donc d'après (1.1) :

$$\Delta > \psi_1(1) + \frac{nr^2}{10}, \quad \forall k \in \mathbb{N}, \quad (1.2)$$

d'autre part, du lemme **2.6.3** on a :

$$f(x^k) - f(x^{k-1}) = -\Delta,$$

d'où de (1.2) on a :

$$f(x^i) - f(x^{i-1}) < - \left(\psi_1(1) + \frac{nr^2}{10} \right),$$

et par suite

$$\sum_{i=1}^k (f(x^i) - f(x^{i-1})) < - \sum_{i=1}^k \left(\psi_1(1) + \frac{nr^2}{10} \right),$$

ce qui donne :

$$f(x^k) - f(x^0) < -k \left(\psi_1(1) + \frac{nr^2}{10} \right), \quad x^0 = \frac{e_n}{n},$$

donc :

$$f(x^k) < -k \left(\psi_1(1) + \frac{nr^2}{10} \right) + f(x^0),$$

comme :

$$f(x^0) = f\left(\frac{e_n}{n}\right) = n \log c^t \frac{e_n}{n} - \sum_{i=1}^n \log \frac{1}{n} = n \log c^t e_n,$$

on a :

$$f(x^k) < -k \left(\psi_1(1) + \frac{nr^2}{10} \right) + n \log c^t e_n,$$

ou aussi :

$$\frac{f(x^k)}{n} < \frac{-k \left(\psi_1(1) + \frac{nr^2}{10} \right) + n \log c^t e_n}{n}, \quad (1.3)$$

d'après le lemme **2.6.2** on a :

$$c^t x^k < \exp \left(\frac{f(x^k)}{n} \right).$$

Rappelons que l'algorithme converge lorsque $c^t x^k$ soit inférieur à ϵ ($\epsilon > 0$ assez petit). Il suffit donc de chercher k qui vérifie :

$$\exp \left(\frac{-k \left(\psi_1(1) + \frac{nr^2}{10} \right) + n \log c^t e_n}{n} \right) < \epsilon, \quad (1.4)$$

ce qui revient à trouver k vérifiant :

$$\left(\frac{-k \left(\psi_1(1) + \frac{nr^2}{10} \right) + n \log c^t e_n}{n} \right) < \log \epsilon, \quad (1.5)$$

comme $\psi_1(1) = 1 - \log 2$, l'inégalité (1.5) se traduit par :

Si

$$k > \frac{n}{\left(1 - \log 2 + \frac{nr^2}{10}\right)} \log \frac{c^t e_n}{\epsilon},$$

alors $c^t x^k \leq \epsilon$ d'où le résultat. ■

Remarque 2.6.2 :

- D'après la preuve du lemme **2.6.5 b**), on sait que

$$\varphi(\alpha_M) = \alpha_M n^2 r^2 + \frac{1}{5} \alpha_M n^3 r^4 - \psi_1(-\alpha_M n r) < \Delta$$

On montre facilement que :

$$\alpha_M = \arg \max_{\left[0, \frac{1}{nr}\right]} \varphi(\alpha) = \frac{5 + nr^2}{5 + (5 + nr^2)nr},$$

et que

$$\varphi(\alpha_M) = \max_{[0, \frac{1}{nr}]} \varphi(\alpha) = \psi_1 \left(\frac{1}{5} nr (5 + nr^2) \right),$$

En effet, d'après le lemme **2.6.5 b)** on a :

$$\Delta > \psi_1 \left(\frac{1}{5} nr (5 + nr^2) \right) = \varphi(\alpha_M) > \varphi(\alpha),$$

i.e., :

$$\Delta > \varphi(\alpha), \text{ tel que } \alpha \in]0, \frac{1}{nr}[$$

on conclut que pour $\alpha = \alpha_M$ on a une réduction maximale de $f(x^k)$ (car $f(x^k) - f(x^{k-1}) = -\Delta$).

2.6.2 Deuxième amélioration du pas de déplacement

Dans cette partie, en se basant sur les travaux Schrijver [47] et Padberg [37], nous fournissons dans le lemme suivant un nouveau pas de déplacement α_B , $\frac{1}{2} < \alpha_B < 1$, qui permet de réduire le nombre d'itérations nécessaires pour la convergence de l'algorithme de Karmarkar.

Lemme 2.6.6 ([12]) pour :

$$\alpha_B = \frac{n}{2n-1},$$

L'algorithme de Karmarkar converge après $\frac{n}{1-\log 2 + \frac{nr^2}{2}} \log \left(\frac{c^t \epsilon_n}{\epsilon} \right)$ itérations.

Remarque 2.6.3 :

- Par définition : $\frac{1}{2} < \alpha_B$
- On peut voir facilement que : $\psi_1 \left(1 + \frac{1}{n-1} \right) \geq 1 - \log 2 + \frac{nr^2}{2}$

du fait que

$$\psi_1 \left(1 + \frac{1}{n-1} \right) = 1 + \frac{1}{n-1} - \log \left(2 + \frac{1}{n-1} \right)$$

et comme : $\log(2+x) - \log 2 \leq \frac{1}{2}x$ pour $0 < x < 1$ alors :

$$\psi_1 \left(1 + \frac{1}{n-1} \right) \geq 1 + \frac{1}{n-1} - \frac{1}{2(n-1)} - \log 2$$

et nous avons : $\frac{1}{n-1} = nr^2$, on a :

$$\psi_1 \left(1 + \frac{1}{n-1} \right) \geq 1 - \log 2 + \frac{nr^2}{2}$$

• Pour prouver le lemme **2.6.6**, on a besoin du lemme suivant :

Lemme 2.6.7 ([12]) Pour $0 < \alpha < 1$ et y^k défini par l'algorithme de Karmarkar, nous avons :

a) $\sum_{i=1}^n \log(1 - \alpha n r d_i^k) \geq (n-1) \log \left(1 + \frac{\alpha}{n-1} \right) + \log(1 - \alpha).$

b) Pour $\alpha = \alpha_B = \frac{n}{2n-1}$, on a : $\Delta \geq 1 - \log 2 + \frac{nr^2}{2}$ qui représente la borne inférieure des valeurs décroissantes potentiels à chaque itération.

Preuve. a) D'après Lemme **2.5.3**, on a :

$$\frac{1}{n \left(\prod_{i=1}^n y_i^k \right)^{\frac{1}{n}}} \leq \frac{1}{1 + \frac{\alpha}{n-1}} \left(\frac{1 + \frac{\alpha}{n-1}}{1 - \alpha} \right)^{\frac{1}{n}}, \quad n \geq 3$$

donc :

$$\begin{aligned} \prod_{i=1}^n y_i^k &\geq \frac{1}{n^n} \left(1 + \frac{\alpha}{n-1} \right)^{(n-1)} (1 - \alpha) \\ \Leftrightarrow \sum_{i=1}^n \log y_i^k &\geq -n \log n + (n-1) \log \left(1 + \frac{\alpha}{n-1} \right) + \log(1 - \alpha) \\ \Leftrightarrow \sum_{i=1}^n \log(1 - \alpha n r d_i^k) &\geq (n-1) \log \left(1 + \frac{\alpha}{n-1} \right) + \log(1 - \alpha) \end{aligned}$$

d'où le résultat.

b) Il est connu que l'itération de Karmarkar est définie par :

$$y^k = \frac{e_n}{n} - \alpha r d^k$$

donc :

$$e_n^t d^k = 0, \text{ (parce que } y^k \in S_n)$$

D'après Lemme **2.5.1**, on a :

$$\frac{c^t D_k y^k}{c^t D_k e_n} = \frac{1}{n} \left(\frac{c^t D_k y^k}{c^t D_k \frac{e_n}{n}} \right) \leq \frac{1}{n} \left(1 - \alpha \frac{r}{R} \right) = \frac{1}{n} \left(1 - \alpha \frac{1}{n-1} \right) \quad (2.1)$$

et de Lemme **2.6.3**, on a :

$$\Delta = n \log \frac{c^t D_k e_n}{c^t D_k y^k} + \sum_{i=1}^n \log y_i^k$$

puis en utilisant l'inégalité (2.1), on a :

$$\begin{aligned} \Delta &\geq -n \log \frac{1}{n} \left(1 - \alpha \frac{1}{n-1} \right) + \sum_{i=1}^n \log y_i^k \\ &\geq \left[-n \log \left(1 - \alpha \frac{1}{n-1} \right) + n \log n \right] + \left[-n \log n + \sum_{i=1}^n \log \left(1 - \alpha n r d_i^k \right) \right] \end{aligned}$$

D'après Lemme **2.6.7 a)**, on a :

$$\sum_{i=1}^n \log \left(1 - \alpha n r d_i^k \right) \geq (n-1) \log \left(1 + \frac{\alpha}{n-1} \right) + \log (1 - \alpha)$$

donc :

$$\Delta \geq (n-1) \log \left(\frac{1 + \frac{\alpha}{n-1}}{1 - \frac{\alpha}{n-1}} \right) - \log \left(1 - \alpha \frac{1}{n-1} \right) + \log (1 - \alpha) \quad (2.2)$$

mais pour $|x| < 1$, on a :

$$\log \left(\frac{1+x}{1-x} \right) \geq 2x \text{ et } -\log (1-x) \geq x$$

l'inégalité (2.2) devient :

$$\Delta \geq \left(2 + \frac{1}{n-1}\right) \alpha + \log(1 - \alpha) \quad (2.3)$$

Pour $\alpha_B = \frac{n}{2n-1}$, l'inégalité (2.3) devient :

$$\Delta \geq \psi_1 \left(1 + \frac{1}{n-1}\right)$$

D'après la remarque **2.6.3**, on a :

$$\Delta \geq 1 - \log 2 + \frac{nr^2}{2}$$

Preuve du lemme 2.6.6 :

Comme $\Delta = f(x^{k-1}) - f(x^k)$, alors :

$$f(x^{k-1}) - f(x^k) \geq 1 - \log 2 + \frac{nr^2}{2}, \forall k \in \mathbb{N}^*, \quad (2.4)$$

par conséquent :

$$\sum_{i=1}^k (f(x^i) - f(x^{i-1})) \leq - \sum_{i=1}^k \left(1 - \log 2 + \frac{nr^2}{2}\right),$$

ce qui donne :

$$f(x^k) - f(x^0) \leq -k \left(1 - \log 2 + \frac{nr^2}{2}\right), \text{ tel que } x^0 = \frac{e_n}{n},$$

donc :

$$f(x^k) \leq -k \left(1 - \log 2 + \frac{nr^2}{2}\right) + f(x^0),$$

telle que :

$$f(x^0) = f\left(\frac{e_n}{n}\right) = n \log c^t \frac{e_n}{n} - \sum_{i=1}^n \log \frac{1}{n} = n \log c^t e_n,$$

ensuite, nous avons :

$$f(x^k) \leq -k \left(1 - \log 2 + \frac{nr^2}{2} \right) + n \log c^t e_n,$$

ou aussi bien :

$$\frac{f(x^k)}{n} \leq \frac{-k \left(1 - \log 2 + \frac{nr^2}{2} \right) + n \log c^t e_n}{n}. \quad (2.5)$$

d'après lemme **2.6.2**, on a :

$$c^t x^k < \exp \left(\frac{f(x^k)}{n} \right).$$

Rappelons que l'algorithme converge lorsque $c^t x^k$ est inférieur à ϵ ($\epsilon > 0$ assez petit). Il suffit de chercher k vérifie :

$$\exp \left(\frac{-k \left(1 - \log 2 + \frac{nr^2}{2} \right) + n \log c^t e_n}{n} \right) < \epsilon,$$

ce qui donne :

$$\left(\frac{-k \left(1 - \log 2 + \frac{nr^2}{2} \right) + n \log c^t e_n}{n} \right) < \log \epsilon, \quad (2.6)$$

L'inégalité (2.6) conduit à :

$$-k \left(1 - \log 2 + \frac{nr^2}{2} \right) < n \log \frac{\epsilon}{c^t e_n},$$

Et enfin, si :

$$k > \frac{n}{1 - \log 2 + \frac{nr^2}{2}} \log \left(\frac{c^t e_n}{\epsilon} \right)$$

alors $c^t x^k \leq \epsilon$. d'où le résultat. ■

2.7 Expérimentations numériques

Nous présentons dans cette partie des tests comparatifs sur différents exemples numériques tirés de la littérature [28]. Nous avons appliqué trois variantes de l'algorithme de Ye-Lustig (cas z^* est inconnu) en utilisant trois pas de déplacement : le pas de déplacement α_S de Schrijver, le nouveau pas déplacement α_M de (Bouafia et Benterki), puis finalement le nouveau pas déplacement α_B de (Bouafia, Benterki & Yassine). La précision ϵ est pris entre 10^{-6} et 10^{-4} .

Dans chaque cas, k représente le nombre d'itérations nécessaires pour trouver la solution optimale.

2.7.1 Exemple avec taille fixe

On présente dans cette partie des tests numériques comparatifs effectués sur différents exemples de programmes linéaires mis sous la forme standard suivante :

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

Nous avons testé les trois variantes de l'algorithme de Ye-Lustig en utilisant les trois pas α_S , α_M et α_B .

Dans chaque cas, on note par :

α_S , α_M , α_B : Les pas de déplacement utilisés dans chaque algorithme.

k : Le nombre d'itérations nécessaires pour l'optimalité.

T : Le temps d'exécution nécessaire pour l'optimalité ; exprimé en secondes.

Exemple 1 $A = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$, $b = (8, 7, 3)^t$, $c = (-4, -5, 0, 0, 0)^t$

a) Alternative de Schrijver :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (2.2535, 1.5744, 1.9186, 1.5977, 1.4256)^t$$

- Solution optimale trouvée :

$$x^* = (3.0000, 2.0000, 0.0000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_S = 0.477$

- Nombre d'itérations : $k = 26$

- Valeur optimale trouvée : $z^* = -22.0000$

b) Alternative de (Bouafia & Benterki) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (2.2535, 1.5744, 1.9187, 1.5977, 1.4256)^t$$

- Solution optimale trouvée :

$$x^* = (3.0000, 2.0000, 0.0000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_M = 0.486$

- Nombre d'itérations : $k = 25$

- Valeur optimale trouvée : $z^* = -22.0000$

c) Alternative de (Bouafia, Benterki & Yassine) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (2.2531, 1.5743, 1.9191, 1.5980, 1.4256)^t$$

- Solution optimale trouvée :

$$x^* = (3.0000, 2.0000, 0.0000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_B = 0.538$

- Nombre d'itérations : $k = 24$

- Valeur optimale trouvée : $z^* = -22.0000$

Example 2 $A = \begin{pmatrix} 0 & 1 & 2 & -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 1 & 2 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & -1 & -2 & 0 & 0 & 0 & 1 \\ 1 & 3 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$

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

a) Alternative de Schrijver :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (0.2872, 0.1764, 0.0863, 0.2242, 0.3902, 0.4851, 0.5946, 2.4551, 1.3647)^t$$

- Solution optimale trouvée :

$$x^* = (0.0000, 0.0000, 0.5000, 0.0000, 0.0000, 0.0000, 0.5000, 4.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_S = 0.487$

- Nombre d'itérations : $k = 47$

- Valeur optimale trouvée : $z^* = -1.0000$

b) Alternative de (Bouafia & Benterki) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (0.2872, 0.1764, 0.0863, 0.2242, 0.3902, 0.4851, 0.5946, 2.4551, 1.3647)^t$$

- Solution optimale trouvée :

$$x^* = (0.0000, 0.0000, 0.5000, 0.0000, 0.0000, 0.0000, 0.5000, 4.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_M = 0.492$

- Nombre d'itérations : $k = 46$

- Valeur optimale trouvée : $z^* = -1.0000$

c) Alternative de (Bouafia, Benterki & Yassine) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (0.2873, 0.1764, 0.0862, 0.2244, 0.3906, 0.4852, 0.5949, 2.4540, 1.3653)^t$$

- Solution optimale trouvée :

$$x^* = (0.0000, 0.0000, 0.5000, 0.0000, 0.0000, 0.0000, 0.5000, 4.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_B = 0.524$

- Nombre d'itérations : $k = 36$

- Valeur optimale trouvée : $z^* = -1.0000$

Example 3 $A = \begin{pmatrix} 2 & -6 & 2 & 7 & 3 & 8 & 1 & 0 & 0 & 0 & 0 \\ -3 & -1 & 4 & -3 & 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 8 & -3 & 5 & -2 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 8 & 7 & -1 & 3 & 0 & 0 & 0 & 1 & 0 \\ 5 & 2 & -3 & 6 & -2 & -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$

$b = (1, -2, 4, 1, 5)^t$, $c = (-18, 7, -12, -5, 0, -8, 0, 0, 0, 0, 0)^t$

a) Alternative de Schrijver :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (2.2654, 7.0153, 0.0811, 0.1630, 11.0981, 0.1712, 2.5943, 0.5355, 6.5011, 0.7328, 1.2751)^t$$

- Solution optimale trouvée :

$$x^* = (2.0000, 4.0001, 0.0000, 7.0001, 0.0000, 0.0000, 1.0000, 0.0000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_S = 0.487$

- Nombre d'itérations : $k = 46$

- Valeur optimale trouvée : $z^* = -8.0000$

b) Alternative de (Bouafia & Benterki) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (2.2654, 7.0153, 0.0811, 0.1630, 11.0981, 0.1712, 2.5943, 0.5355, 6.5011, 0.7328, 1.2751)^t$$

- Solution optimale trouvée :

$$x^* = (2.0000, 4.0001, 0.0000, 7.0001, 0.0000, 0.0000, 1.0000, 0.0000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_M = 0.493$

- Nombre d'itérations : $k = 46$
- Valeur optimale trouvée : $z^* = -8.0000$

c) Alternative de (Bouafia, Benterki & Yassine) :

- Solution strictement réalisable initiale trouvée :
- $$x^0 = (2.2640, 7.0101, 0.0811, 0.1629, 11.0892, 0.1711, 2.5949, 0.5354, 6.4973, 0.7328, 1.2753)^t$$
- Solution optimale trouvée :
- $$x^* = (2.0003, 4.0008, 0.0000, 7.0003, 0.0000, 0.0000, 1.0004, 0.0000, 0.0000, 0.9996)^t$$
- Pas de déplacement : $\alpha_B = 0.520$
 - Nombre d'itérations : $k = 35$
 - Valeur optimale trouvée : $z^* = -8.0000$

Example 4 $A = \begin{pmatrix} 1 & 0 & -4 & 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 3 & 1 & 0 & -1 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & 5 & -3 & 3 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 & 1 & -5 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2 & 1 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & -3 & 2 & -1 & 4 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$

$b = (1, 4, 4, 5, 7, 5)^t$, $c = (-4, -5, -1, -3, 5, -8, 0, 0, 0, 0, 0, 0)^t$

a) Alternative de Schrijver :

- Solution strictement réalisable initiale trouvée :
- $$x^0 = (0.1546, 0.4468, 1.6114, 1.9394, 0.6863, 0.1695, 0.6169, 0.4530, 2.7394, 1.7291, 1.5999, 1.1549)^t$$
- Solution optimale trouvée :
- $$x^* = (0.0000, 0.0000, 2.5000, 3.5000, 0.0000, 0.5000, 0.0000, 0.0000, 0.5000, 0.5000, 0.0000, 1.0000)^t$$
- Pas de déplacement : $\alpha_S = 0.490$
 - Nombre d'itérations : $k = 48$

- Valeur optimale trouvée : $z^* = -17.0000$

b) Alternative de (Bouafia & Benterki) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (0.1546, 0.4468, 1.6114, 1.9394, 0.6863, 0.1695, 0.6169, 0.4530, 2.7394, 1.7291, 1.5999, 1.1549)^t$$

- Solution optimale trouvée :

$$x^* = (0.0000, 0.0000, 2.5000, 3.5000, 0.0000, 0.5000, 0.0000, 0.0000, 0.5000, 0.5000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_M = 0.494$

- Nombre d'itérations : $k = 47$

- Valeur optimale trouvée : $z^* = -17.0000$

c) Alternative de (Bouafia, Benterki & Yassine) :

- Solution strictement réalisable initiale trouvée :

$$x^0 = (0.1546, 0.4470, 1.6113, 1.9392, 0.6864, 0.1695, 0.6171, 0.4531, 2.7391, 1.7291, 1.5998, 1.1555)^t$$

- Solution optimale trouvée :

$$x^* = (0.0000, 0.0000, 2.5000, 3.5000, 0.0000, 0.5000, 0.0000, 0.0000, 0.5000, 0.5000, 0.0000, 1.0000)^t$$

- Pas de déplacement : $\alpha_B = 0.519$

- Nombre d'itérations : $k = 44$

- Valeur optimale trouvée : $z^* = -17.0000$

Example 5 : $A =$

$$\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix},$$

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

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

a) Alternative de Schrijver :

- Solution strictement réalisable initiale trouvée :

$$x^0 = \begin{bmatrix} 3.5855, 0.8285, 1.5283, 0.3365, 0.1905, 1.3296, 0.6197, 1.5948, 0.8395, 2.3755, 1.7512, \\ 1.3805, 0.5419, 0.7679, 3.5855, 0.8148, 1.7514, 0.5653, 1.1981, 0.0902, 1.1162, 1.3134, \\ 1.1058, 2.1118, 0.7679 \end{bmatrix}^t$$

- Solution optimale trouvée :

$$x^* = \begin{bmatrix} 0.0000, 1.0000, 2.0000, 0.0000, 0.5000, 0.1448, 0.0000, 0.0000, 0.0000, 1.5000, 0.0000, \\ 3.7500, 0.1208, 0.0000, 6.0000, 0.0000, 0.0000, 0.0000, 1.7103, 1.1448, 2.0000, 0.00037, \\ 4.5000, 0.00000, 0.0084 \end{bmatrix}^t$$

- Pas de déplacement : $\alpha_S = 0.495$

- Nombre d'itérations : $k = 69$

- Valeur optimale trouvée : $z^* = -13.2500$

b) Alternative de (Bouafia & Benterki) :

Solution strictement réalisable initiale trouvée :

$$x^0 = \begin{bmatrix} 3.0984, 0.9014, 1.4927, 0.3289, 0.1958, 1.3179, 0.6168, 1.5421, 0.8483, 2.3665, 1.7333, \\ 1.4135, 0.5377, 0.7556, 3.0984, 0.7045, 1.8505, 0.5594, 1.1898, 1.0844, 1.0751, 1.2672, \\ 1.0884, 2.0729, 0.7556 \end{bmatrix}^t$$

- Solution optimale trouvée :

$$x^* = \begin{bmatrix} 0.0000, 1.0000, 2.0000, 0.0000, 0.5000, 0.1367, 0.0000, 0.0000, 0.0000, 1.5000, 0.0000, \\ 3.7500, 0.1211, 0.0000, 6.0000, 0.0000, 0.0000, 0.0000, 1.7103, 1.1448, 2.0000, 0.00037, \\ 4.5000, 0.00000, 0084 \end{bmatrix}^t$$

- Pas de déplacement : $\alpha_M = 0.497$

- Nombre d'itérations : $k = 69$

- Valeur optimale trouvée : $z^* = -13.2500$

c) Alternative de (Bouafia, Benterki & Yassine) :

Solution strictement réalisable initiale trouvée :

$$x^0 = \begin{bmatrix} 3.0984, 0.9014, 1.4927, 0.3289, 0.1958, 1.3179, 0.6168, 1.5421, 0.8496, 2.3665, 1.7333, \\ 1.4135, 0.5377, 0.7556, 3.0984, 0.7045, 1.8505, 0.5594, 1.1898, 1.0844, 1.0751, 1.2672, \\ 1.0858, 2.0729, 0.7556 \end{bmatrix}^t$$

- Solution optimale trouvée :

$$x^* = \begin{bmatrix} 0.0000, 1.0000, 2.0000, 0.0000, 0.5000, 0.1367, 0.0000, 0.0000, 0.0000, 1.5000, 0.0000, \\ 500, 0.1211, 0.0000, 6.0000, 0.0000, 0.0000, 0.0000, 1.7103, 1.1448, 2.0000, 0.00037, \\ 4.5000, 0.00000, 0065 \end{bmatrix}^t$$

- Pas de déplacement : $\alpha_B = 0.509$

- Nombre d'itérations : $k = 67$

- Valeur optimale trouvée : $z^* = -13.2500$

tableau comparatif

Exemple	Taille(m, n)	Nombre d'itérations utilisant α_S	Nombre d'itérations utilisant α_M	Nombre d'itérations utilisant α_B
1	(3, 5)	$k = 26$	$k = 25$	24
2	(5, 9)	$k = 47$	$k = 46$	36
3	(5, 11)	$k = 46$	$k = 46$	35
4	(6, 12)	$k = 48$	$k = 47$	44
5	(11, 25)	$k = 69$	$k = 69$	67

2.7.2 Exemple avec taille variable

Exemple cube

$$n = 2m,$$

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

$$c(i) = -1, c(i + m) = 0 \text{ et } b(i) = 2, \text{ pour } i = 1, \dots, m.$$

- Solution réalisable strictement initiale trouvée : $x^0 = (1, 1, \dots, 1)^t$

- La valeur optimale est : $z^* = -2m$.

$$\text{- Solution optimale trouvée : } x_i^* = \begin{cases} 2 & \text{si } i = 1, \dots, m. \\ 0 & \text{si } i = m + 1, \dots, n. \end{cases}$$

tableau comparatif

Taille (m, n)	Alternative utilisant α_S	Alternative utilisant α_M	Alternative utilisant α_B
$(3, 6)$	$k = 24$ $z^* = -5.9998$	$k = 23$ $z^* = -5.9998$	$k = 21$ $z^* = -5.9998$
$(25, 50)$	$k = 62$ $z^* = -49.9933$	$k = 61$ $z^* = -49.9924$	$k = 61$ $z^* = -49.9933$
$(50, 100)$	$k = 83$ $z^* = -99.9775$ $T = 0.33s$	$k = 83$ $z^* = -99.9777$ $T = 0.31s$	$k = 83$ $z^* = -99.9789$ $T = 0.31s$
$(100, 200)$	$k = 113$ $z^* = -199.9369$ $T = 3.90s$	$k = 113$ $z^* = -199.9371$ $T = 3.89s$	$k = 112$ $z^* = -199.9343$ $T = 3.89s$
$(150, 300)$	$k = 135$ $z^* = -299.8833$ $T = 15.24s$	$k = 135$ $z^* = -299.8836$ $T = 15.19$	$k = 134$ $z^* = -299.8788$ $T = 15.19$
$(200, 400)$	$k = 152$ $z^* = -399.8096$ $T = 39.89s$	$k = 152$ $z^* = -399.8100$ $T = 39.82s$	$k = 152$ $z^* = -399.8123$ $T = 39.82s$
$(250, 500)$	$k = 168$ $z^* = -499.7391$ $T = 84.31s$	$k = 168$ $z^* = -499.7395$ $T = 84.22s$	$k = 168$ $z^* = -499.7420$ $T = 84.22s$

Exemple de Hilbert

$$n = 2m, A[i, j] = \frac{1}{i+j}, \text{ pour } i, j = 1, \dots, m. \text{ Et}$$

$$A[i, i+m] = 1, b[i] = \sum_{j=1}^m \frac{1}{i+j}, c[i] = b[i] + \frac{1}{(i+1)}, c[i+m] = 0, \text{ pour } i = 1, \dots, m.$$

Taille (m, n)	Alternative utilisant α_S	Alternative utilisant α_M	Alternative utilisant α_B
$(4, 8)$	$k = 40$ $z^* = 0.0000$	$k = 39$ $z^* = 0.0000$	$k = 36$ $z^* = 0.0000$
$(25, 50)$	$k = 95$ $z^* = 0.0001$	$k = 94$ $z^* = 0.0001$	$k = 93$ $z^* = 0.0001$
$(50, 100)$	$k = 130$ $z^* = 0.0002$ $T = 0.5s$	$k = 130$ $z^* = 0.0002$ $T = 0.5s$	$k = 129$ $z^* = 0.0002$ $T = 0.5s$
$(100, 200)$	$k = 180$ $z^* = 0.0005$ $T = 6.19s$	$k = 180$ $z^* = 0.0005$ $T = 6.17s$	$k = 179$ $z^* = 0.0005$ $T = 6.17s$
$(150, 300)$	$k = 217$ $z^* = 0.0010$ $T = 24.70s$	$k = 217$ $z^* = 0.0010$ $T = 24.68s$	$k = 216$ $z^* = 0.0010$ $T = 24.68s$
$(200, 400)$	$k = 247$ $z^* = 0.0016$ $T = 65.94s$	$k = 247$ $z^* = 0.0016$ $T = 65.91s$	$k = 247$ $z^* = 0.0016$ $T = 65.91s$
$(250, 500)$	$k = 274$ $z^* = 0.0023$ $T = 142.17s$	$k = 274$ $z^* = 0.0023$ $T = 142.11s$	$k = 274$ $z^* = 0.0023$ $T = 142.11s$

Commentaries : Les résultats numériques montrent que le nombre d'itérations de notre approche est toujours inférieur ou égal à celui de l'approche de Sherijver. Ceci consolide et confirme notre objectif théorique.

2.8 Conclusion

Dans ce chapitre, nous avons apporté des contributions d'ordre théorique et numérique. En effet, l'introduction de deux nouveaux pas de déplacement (α_M, α_B) a impliqué une amélioration de taux de convergence de l'algorithme de Karmarkar. Ce phénomène devient plus intéressant puisque la dimension du problème est prise en compte. Nous avons confirmé les propos théoriques du fait qu'on remarque que nos deux pas (α_M, α_B) sont les itérés de deux suites réelles adjacentes et encadrées (majorées et minorées) par le pas de déplacement de Padberg $\alpha_P = \frac{1}{2}$. En plus, lorsque n tend vers l'infini, les trois suites réelles de α_S , α_M et α_B convergent vers le pas de déplacement de Padberg $\alpha_P = \frac{1}{2}$. Il serait probablement plus intéressant d'insister beaucoup plus sur la forme analytique de la fonction potentiel tout en conservant ses bonnes propriétés dans le but de ramener le pas de déplacement au delà de 1. C'est d'ailleurs la première idée présentée par Padberg. Cette idée a été poursuivie par les travaux de Naseri et al. [35]. Malheureusement, nous avons détecté durant notre recherche que les résultats théoriques obtenus par ces derniers ne sont pas rigoureux et méritent plus de soins malgré leurs succès numériques.

Chapitre 3

Méthode de Trajectoire Centrale (TC) pour la programmation linéaire

3.1 Introduction

Les méthodes de trajectoire centrale (**TC**) ont été introduites à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles possèdent de bonnes propriétés théoriques : une complexité polynomiale et une convergence super-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, en utilisant des nouvelles techniques.

3.2 Méthode de Trajectoire Centrale classique

3.2.1 Présentation de la méthode

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

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

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 3.2.1 *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).

3.2.2 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 :

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

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 :

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

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. [26] ont introduit le paramètre poids, associé à la fonction barrière.

3.3 Méthode de Trajectoire Centrale avec poids

3.3.1 Présentation de la méthode

On considère le problème linéaire sous forme standard suivant :

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

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 :

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

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**

$k = 0, (x^0, y^0, s^0) \in \mathcal{F}_{(PL)}^0 \times \mathcal{F}_{(DL)}^0$ (voir annexes **2-3**, pages **130-132**) ;

Un paramètre de précision $\epsilon > 0$;

$\varrho = \frac{\|X^0 S^0 e\|}{\sqrt{n}}, r = \frac{X^0 S^0 e}{\varrho}, \eta = \min\{r_i\}, \theta = \left(1 - \frac{\sqrt{2}}{2}\right) \eta, \beta = 1 - \frac{\theta}{\sqrt{n}}$ 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 (voir annexe **5**, page **133**)

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 (voir annexe **5**, pages **133**)

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.

3.3.2 Etude de la convergence

L'étude de la convergence a été établie par Z. Kebbiche et al. [26] 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 3.3.1 [26] 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) $\frac{x^{+t} R s^+}{n} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) \frac{x^t R s}{n}$.

3.3.3 Conclusion

Dans ces deux sections précédentes, nous avons donné des études d'ordre théorique et algorithmique sur la méthode de trajectoire centrale. En effet, la méthode de trajectoire centrale avec poids représente une généralisation de la méthode classique et écarte le problème d'initialisation. Cette généralisation a pour but d'assurer la vérification que le nouvel itéré (x^+, y^+, s^+) , solution du système $(S_{\mu r})$, est voisin de la trajectoire centrale $T_r(\theta)$. Cette idée été posé par les travaux de Z. Kebbiche et al. [26]. L'inconvénient de cette approche est la détermination du paramètre θ dans le théorème précédent. En effet, jusqu'à présent on utilise une seule valeur pour θ . Nous n'avons aucune garantie que cette valeur est la meilleure.

Dans la section suivante, nous allons apporter des contributions réelles sur la nouvelle méthode de trajectoire centrale via une fonction noyau. Cette approche a l'avantage de démarrer avec n'importe quel point de départ (x^0, y^0, s^0) qui vérifie le système (S_μ) .

3.4 Méthode de Trajectoire Centrale via une fonction noyau

3.4.1 Présentation de la méthode

Dans ce paragraphe, nous considérons le problème d'optimisation linéaire primal donné par :

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

où $A \in \mathbb{R}^{m \times n}$, $\text{rang}(A) = m$, $b \in \mathbb{R}^m$, et $c, x \in \mathbb{R}^n$, et son problème dual associé :

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

En 1984, Karmarkar [25] a proposé une nouvelle méthode en temps polynomial pour résoudre des programmes linéaires. Cette méthode, et ses variantes, développées par la suite, s'appellent Méthodes de Points Intérieurs, notées **IPMs**. Les lecteurs peuvent se référer à des références de base traitant le sujet, voir Y. Q. Bai et al. [5], et J. Peng et al. [40], C. Roos et al. [45], Y. Ye [56]. Sans perte de généralité, nous supposons que (P) et (D) satisfont la Condition de Points Intérieurs, notée **IPC**, i.e., il existe (x^0, y^0, s^0) tel que

$$Ax^0 = b, x^0 > 0, A^t y^0 + s^0 = c, s^0 > 0. \quad (\text{voir annexes } \mathbf{2-3}, \text{ pages } \mathbf{130-132}) \quad (3.1)$$

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

$$\begin{cases} Ax = b, & x \geq 0, \\ A^t y + s = c, & s \geq 0, \\ xs = 0. \end{cases} \quad (3.2)$$

Le système (3.2) est obtenu par le théorème de la dualité forte appliqué aux problèmes duaux (P) et (D) . L'idée principale d'**IPMs** est de remplacer la troisième équation du système (3.2), dite condition de complémentarité, par l'équation paramétrée $xs = \mu e$, avec $\mu > 0$. Nous obtenons le système suivant :

$$\begin{cases} Ax = b, x \geq 0, \\ A^t y + s = c, s \geq 0, \\ xs = \mu e. \end{cases} \quad (3.3)$$

Nous remarquons que ce système est le même système (SN_μ) présenté auparavant, ce qui prouve aussi que c'est une méthode barrière.

Assez surprenant, si la condition **IPC** est satisfaite, alors il existe une solution $(x(\mu), y(\mu), s(\mu))$ du dernier système, pour chaque $\mu > 0$, et cette solution est unique. Nous appelons $x(\mu)$ le μ -centre de (P) et $(y(\mu), s(\mu))$ le μ -centre de (D) . L'ensemble de μ -centres (avec μ prenant plusieurs valeurs décroissantes des nombres réels positifs qui convergent vers 0) forme un trajet, appelé la trajectoire centrale de (P) et (D) .

L'importance de la trajectoire centrale pour l'optimisation linéaire a été reconnue par plusieurs chercheurs, tels que Sonnevend [50] et Megiddo [34]. Si $\mu \rightarrow 0$, la limite de la trajectoire centrale existe, et étant donné que les points limites satisfont la condition de complémentarité, la limite donne des solutions optimales de (P) et (D) .

D'un point de vue théorique, et sans perte de généralité, on peut supposer que l'**IPC** est satisfaite. Pour simplifier les contributions théoriques, et toujours sans perte de généralité, nous pouvons supposer que $x^0 = s^0 = e$.

Nous supposons que $(x(\mu), y(\mu), s(\mu))$ est connu pour certaines valeurs positives de μ . Par exemple, en raison de l'hypothèse ci-dessus, nous pouvons supposer que pour $\mu = 1$, $x(1) = s(1) = e$. Nous réduisons μ à $\mu = (1 - \theta)\mu$ pour un nombre fixé $\theta \in]0, 1[$, et on

résout le système de Newton suivant :

$$\begin{aligned} A\Delta x &= 0, \\ A^t\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs. \end{aligned} \tag{3.4}$$

Le système (3.4) admet une solution unique désignée par $(\Delta x, \Delta y, \Delta s)$. Cette solution est dite la direction de Newton classique pour l'optimisation linéaire. Ensuite, μ est à nouveau réduite par le facteur $(1 - \theta)$, et nous appliquons de nouveau la méthode de Newton ciblant les nouveaux μ -centres, et ainsi de suite. Ce processus est répété jusqu'à ce que μ devient assez petit (obtention de la condition $n\mu \leq \epsilon$), alors dans ce cas nous obtenons une ϵ -solution des problèmes (P) et (D) . Le nouvel itéré de la méthode de Newton avec un pas fixé est donné par :

$$x_+ = x + \alpha\Delta x, \quad y_+ = y + \alpha\Delta y, \quad s_+ = s + \alpha\Delta s; \text{ où } \alpha \in]0, 1]. \tag{3.5}$$

Maintenant, nous introduisons le vecteur réduit v et les directions de recherche réduite d_x et d_s comme suit :

$$v = \sqrt{\frac{xs}{\mu}}, \quad d_x = \frac{v\Delta x}{x}, \quad d_s = \frac{v\Delta s}{s}. \tag{3.6}$$

Le système (3.4) peut être réécrit comme suit :

$$\begin{cases} A\Delta x = 0, \\ A^t\Delta y + \Delta s = 0, \\ s\Delta x + x\Delta s = -\mu v (v - v^{-1}). \end{cases} \tag{3.7}$$

Qui est équivalent à

$$\begin{cases} \overline{A}d_x = 0, \\ \overline{A}^t\Delta y + d_s = 0, \\ d_x + d_s = v^{-1} - v. \end{cases} \tag{3.8}$$

où $\overline{A} = \frac{1}{\mu}AV^{-1}X$, $V = \text{diag}(v)$, $X = \text{diag}(x)$. A noter que la partie droite de la troisième équation du système (3.8) n'est autre que l'opposé du gradient de la fonction barrière logarithmique $\Phi(v)$, i.e., $d_x + d_s = -\nabla\Phi(v)$, alors le système (3.8) peut être réécrit comme suit :

$$\begin{cases} \overline{A}d_x = 0, \\ \overline{A}^t \Delta y + d_s = 0, \\ d_x + d_s = -\nabla\Phi(v). \end{cases} \quad (3.9)$$

où la fonction barrière $\Phi(v) : \mathbb{R}_{++}^n \rightarrow \mathbb{R}_+$ est définie comme suit :

$$\Phi(v) = \Phi(x, s; \mu) = \sum_{i=1}^n \psi(v_i), \quad (3.10)$$

avec

$$\psi(v_i) = \frac{v_i^2 - 1}{2} - \log v_i. \quad (3.11)$$

Remarque 3.4.1 Si $(d_x, \Delta y, d_s)$ est une solution du système (3.9), alors d_x, d_s sont orthogonaux.

Nous utilisons $\Phi(v)$ la fonction de proximité pour mesurer la distance entre l'itéré μ -centre et 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 :

$$\delta(v) = \frac{1}{2} \|\nabla\Phi(v)\| = \frac{1}{2} \|d_x + d_s\|. \quad (3.12)$$

Nous appelons $\psi(t)$ la fonction noyau de la fonction barrière logarithmique $\Phi(v)$. A noter que la paire (y, s) coïncide avec le μ -centre $(y(\mu), s(\mu))$ si et seulement si $v = e$. On peut facilement vérifier que la fonction noyau $\psi(t)$, définie par (3.11), est une fonction strictement convexe, qui est définie pour chaque $t \in \mathbb{R}_{++}$ et qui admet un minimum pour $t = 1$, dont la valeur de sa fonction est égale à 0. Il ressort clairement de la description précédente que la proximité de (x, s) à $(x(\mu), s(\mu))$ est mesurée par la valeur de $\Phi(v)$ et avec $\tau > 0$ en tant que valeur de seuil. Si $\Phi(v) \leq \tau$, alors nous commençons une

nouvelle itération externe en effectuant un μ -mise à jour, dans le cas contraire ($\Phi(v) > \tau$), nous entrons dans une itération interne par le calcul des directions de recherche à l'itération en cours par rapport à la valeur actuelle de μ et nous appliquons (3.5) pour obtenir de nouveaux itérés. Nous répétons la procédure jusqu'à ce qu'on trouve un itéré au voisinage de $(x(\mu), s(\mu))$. Ensuite, μ est à nouveau réduit par le facteur $(1 - \theta)$ avec $0 < \theta < 1$, et nous appliquons la méthode de Newton ciblant les nouveaux μ -centres, et ainsi de suite. Ce processus est répété jusqu'à ce que μ est suffisamment petit, c'est-à-dire jusqu'à ce que $n\mu \leq \epsilon$. A ce stade, nous avons trouvé une solution de ϵ -approximative de l'optimisation linéaire. Les paramètres τ , θ et le pas 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 petit que possible.

Algorithme 3. Générique Primal-dual **IPMs** pour l'optimisation linéaire

Données :

Une fonction de proximité $\Phi(v)$;

Un paramètre de limite $\tau > 0$;

Un paramètre de précision $\epsilon > 0$;

Un paramètre de mise à jour fixe $\theta, 0 < \theta < 1$;

début :

Initialisation : soit (x^0, y^0, s^0) vérifie la **IPC** :

$k = 0$; $\mu^0 = 1$; $v^0 = \sqrt{\frac{x^0 s^0}{\mu^0}}$ (voir annexes **2-3**, pages **130-132**);

Tant que : $n\mu^k > \epsilon$ **faire :**

début (itération externe)

$\mu^{k+1} = (1 - \theta)\mu^k$;

Tant que : $\Phi(v) > \tau$ **faire :**

début (itération interne)

Résoudre le système (3.9) pour déterminer $(d_x, \Delta y, d_s)$ puis $(\Delta x, \Delta y, \Delta s)$;

Calculer le pas de déplacement α (voir section 3.4.6); et poser

$x = x + \alpha \Delta x$;

$y = y + \alpha \Delta y$;

$s = s + \alpha \Delta s$;

$v = \sqrt{\frac{xs}{\mu}}$;

fin (itération interne)

fin (itération externe)

fin.

3.4.2 Fonctions noyau et propriétés

Définition 4, on dit que $\psi : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$ est une fonction noyau si ψ est deux fois différentiable et vérifie les conditions suivantes :

- $\psi'(1) = \psi(1) = 0$,

- $\psi''(t) > 0, \forall t > 0,$
- $\lim_{t \rightarrow 0^+} \psi(t) = \lim_{t \rightarrow +\infty} \psi(t) = +\infty.$

Donc ψ est strictement convexe et minimale en 1 avec $\psi(1) = 0$, et s'écrit comme suit

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\tau) d\tau d\xi,$$

et la dernière condition indique que ψ est une fonction barrière.

Lemme 3.4.1 *Soit ψ une fonction noyau alors :*

- 1- $t\psi'(t) \geq \psi(t), \forall t \geq 1.$
- 2- Si $\psi(t_1) = \psi(t_2)$, tel que $t_1 < 1 < t_2$ alors :

$$\psi'(t_1) < 0, \psi'(t_2) > 0.$$

$$\psi(\beta t_1) < \psi(\beta t_2), \forall \beta > 1.$$

Preuve. 1- On définit $g(t) = t\psi'(t) - \psi(t)$, pour $t \geq 1$. On a :

$$g(1) = 0.$$

$$g'(t) = t\psi''(t) \geq 0.$$

Donc $g(t) \geq 0$, pour $t \geq 1$.

- 2- On suppose que $\psi(t_1) = \psi(t_2)$, avec $t_1 < 1 < t_2$ alors :

$$t_2\psi'(t_2) > \psi(t_2).$$

$$\psi(1) > \psi(t_1) + (1 - t_1)\psi'(t_1).$$

donc :

$$\psi'(t_2) > 0.$$

$$\psi'(t_1) < 0.$$

D'autre part, on définit $f(\beta) = \psi(\beta t_1) - \psi(\beta t_2)$, pour $\beta \geq 1$. On a :

$$f(1) = 0.$$

$$f'(\beta) = t_1 \psi'(\beta t_1) - t_2 \psi'(\beta t_2).$$

comme $\psi''(t) \geq 0$, et $\beta t_1 < \beta t_2$, pour $\beta > 1$, donc :

$$\psi'(\beta t_1) \leq \psi'(\beta t_2), \text{ pour } \beta > 1.$$

D'autre part, $\psi'(\beta t_2) > 0$, pour $t_2 > 1$, pour $\beta > 1$, donc :

$$f'(\beta) < (t_1 - t_2) \psi'(\beta t_2) < 0, \text{ pour } \beta > 1.$$

alors :

$$f(\beta) = \psi(\beta t_1) - \psi(\beta t_2) < 0.$$

Ce qui termine la preuve ■

3.4.3 Qualification d'une fonction noyau

Dans cette partie, nous donnons les conditions de qualification pour une fonction noyau et quelques propriétés concernant ces conditions.

Définition 5 : Une fonction noyau ψ est dite être qualifiée si les propriétés suivantes sont satisfaites :

$$t\psi''(t) + \psi'(t) > 0, \forall t < 1, \tag{3.a}$$

$$t\psi''(t) - \psi'(t) > 0, \forall t > 1, \quad (3.b)$$

$$\psi'''(t) < 0, \forall t > 0, \quad (3.c)$$

$$2[\psi''(t)]^2 - \psi'(t)\psi'''(t) > 0, \forall t < 1, \quad (3.d)$$

$$\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \forall t > 1, \forall \beta > 1. \quad (3.e)$$

3.4.4 Propriétés et relation entre les conditions de qualification

Les trois prochains lemmes sont vérifiés si les trois conditions (3.a), (3.b) et (3.c) sont satisfaites.

Lemme 3.4.2 ([40]) *Soit ψ une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes :*

$$(i) \quad \psi(\sqrt{t_1 t_2}) \leq \frac{(\psi(t_1) + \psi(t_2))}{2}, \text{ pour tous } t_1, t_2 > 0.$$

(ii) la fonction ϕ définie par $\phi(\xi) = \psi(e^\xi)$ est convexe.

$$(iii) \quad t\psi''(t) + \psi'(t) \geq 0, t > 0.$$

Preuve. Signalons que la fonction ϕ est continue donc pour qu'elle soit convexe, il suffit qu'elle soit mid-convexe.

Montrons que (i) \Leftrightarrow (ii)

$$\begin{aligned} \psi(\sqrt{t_1 t_2}) &\leq \frac{(\psi(t_1) + \psi(t_2))}{2}, \forall t_1, t_2 > 0 \\ &\Leftrightarrow \\ \phi\left(\frac{\zeta_1 + \zeta_2}{2}\right) &= \psi(\sqrt{e^{\zeta_1} e^{\zeta_2}}) \leq \frac{(\psi(e^{\zeta_1}) + \psi(e^{\zeta_2}))}{2} \\ &= \frac{\phi(\zeta_1) + \phi(\zeta_2)}{2}, \forall \zeta_1, \zeta_2 \in \mathbb{R} \text{ tel que } t_1 = e^{\zeta_1}, t_2 = e^{\zeta_2} \\ &\Leftrightarrow \phi \text{ est mid-convexe} \end{aligned}$$

Concernant l'équivalence (ii) \Leftrightarrow (iii)

$$\begin{aligned}
\phi \text{ convexe} &\iff \phi''(\xi) = [\psi(e^\xi)]'' = [\psi'(e^\xi) + e^\xi \psi''(e^\xi)] e^\xi \geq 0, \forall \xi \in \mathbb{R} \\
&\iff [t\psi''(t) + \psi'(t)] t \geq 0, \forall t > 0 \text{ tel que } t = e^\xi \\
&\iff t\psi''(t) + \psi'(t) \geq 0, \forall t > 0.
\end{aligned}$$

Ce qui termine la preuve ■

Lemme 3.4.3 *Soit ψ une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes :*

- (i) $\psi(\sqrt{\frac{\zeta_1^2 + \zeta_2^2}{2}}) \leq \frac{\psi(\zeta_1) + \psi(\zeta_2)}{2}, \zeta_1, \zeta_2 > 0.$
- (ii) la fonction φ définie par $\varphi(t) = \psi(\sqrt{t})$ est convexe, $t > 0.$
- (iii) $t\psi''(t) - \psi'(t) \geq 0, t > 0.$

Preuve. Signalons que la fonction φ est continue donc pour qu'elle soit convexe, il suffit qu'elle soit mid-convexe.

Montrons que (i) \Leftrightarrow (ii)

$$\begin{aligned}
\psi(\sqrt{\frac{\zeta_1^2 + \zeta_2^2}{2}}) &\leq \frac{1}{2}(\psi(\zeta_1) + \psi(\zeta_2)), \forall \zeta_1, \zeta_2 \in \mathbb{R}_+^* \\
&\iff \\
\varphi\left(\frac{t_1 + t_2}{2}\right) &= \psi\left(\sqrt{\frac{t_1 + t_2}{2}}\right) \leq \frac{\psi(\sqrt{t_1}) + \psi(\sqrt{t_2})}{2} \\
&= \frac{\varphi(t_1) + \varphi(t_2)}{2}, \forall t_1, t_2 \in \mathbb{R}_+^* \text{ tel que } t_1 = \zeta_1^2, t_2 = \zeta_2^2 \\
&\iff \varphi \text{ est mid-convexe}
\end{aligned}$$

Concernant l'équivalence (ii) \Leftrightarrow (iii)

$$\begin{aligned}
\varphi \text{ convexe} &\iff \varphi''(\xi) = [\psi(\sqrt{t})]'' = \frac{1}{4t^{\frac{3}{2}}}(t\psi''(t) - \psi'(t)) \geq 0, t > 0 \\
&\iff t\psi''(t) - \psi'(t) \geq 0, t > 0
\end{aligned}$$

Ce qui termine la preuve ■

Lemme 3.4.4 ([3]) *Si ψ vérifie (3.b) et (3.c) de la définition 5, alors ψ vérifie (3.e) :*

$$\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \forall t > 1, \forall \beta > 1.$$

Preuve. Supposons que ψ vérifie (3.b) et (3.c) de la définition 5. Soit $t > 1$, on considère

$$f(\beta) = \psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t), \quad \forall \beta > 1,$$

on a

$$f'(\beta) = \psi''(\beta t)[t\psi''(t) - \psi'(t)] - \beta t\psi'(t)\psi'''(\beta t) > 0, \quad \forall \beta > 1,$$

donc la fonction f est strictement croissante et $f(1) = 0$, alors $f(\beta) > 0$ pour tout $\beta > 1$.

Ce qui termine la preuve ■

Lemme 3.4.5 *Soit ψ satisfait (3.c) alors :*

1- ψ admet les propriétés suivantes, en distinguant deux cas différents :

cas 1 : Pour tout $t < 1$, on a :

$$\begin{aligned} \psi(t) &< \frac{\psi'(t)}{2}(t-1). \\ \psi''(t)(t-1) &> \psi'(t). \\ \frac{1}{2}(t-1)^2\psi''(1) &< \psi(t) < \frac{\psi''(t)}{2}(t-1)^2. \end{aligned}$$

cas 2 : Pour tout $t > 1$, on a :

$$\begin{aligned} \frac{\psi'(t)}{2}(t-1) &< \psi(t). \\ \psi''(t)(t-1) &< \psi'(t). \\ \frac{\psi''(t)}{2}(t-1)^2 &< \psi(t) < \frac{\psi''(1)}{2}(t-1)^2. \end{aligned}$$

2- Si $\psi(t_1) = \psi(t_2)$, telle que $t_1 < 1 < t_2$ alors :

$$\psi'(t_2) < -\psi'(t_1).$$

Preuve. 1- On définit $f(t) = 2\psi(t) - (t-1)\psi'(t)$, pour $t > 0$. On a :

$$\begin{aligned} f(1) &= 0, \\ f'(t) &= \psi'(t) - (t-1)\psi''(t), \\ f'(1) &= 0, \\ f''(t) &= -(t-1)\psi'''(t) \end{aligned}$$

donc nous distinguons les deux cas :

cas 1 : si $t < 1$, alors

$$f''(t) = -(t-1)\psi'''(t) < 0,$$

donc

$$\begin{aligned} f'(t) &= \psi'(t) - (t-1)\psi''(t) > 0, \\ f(t) &= 2\psi(t) - (t-1)\psi'(t) < 0, \end{aligned}$$

donc

$$\begin{aligned} \psi'(t) &> (t-1)\psi''(t), \\ \psi(t) &< \frac{1}{2}(t-1)\psi'(t), \end{aligned}$$

En utilisant le développement de Taylor au voisinage de $t_0 = 1$, on a :

$$\begin{aligned} \psi(t) &= \frac{\psi''(1)}{2}(t-1)^2 + \frac{\psi'''(\xi)}{6}(t-1)^3, \quad t < \xi < 1. \\ \psi(t) &= \frac{\psi''(\zeta)}{2}(t-1)^2, \quad t < \zeta < 1. \end{aligned}$$

d'après (3.c) comme $\psi'''(t) < 0$, donc

$$\frac{1}{2}(t-1)^2\psi''(1) < \psi(t) < \frac{\psi''(t)}{2}(t-1)^2,$$

alors :

$$\begin{aligned}\psi(t) &< \frac{\psi'(t)}{2} (t-1) . \\ \psi'(t) &> \psi''(t) (t-1) . \\ \frac{1}{2} (t-1)^2 \psi''(1) &< \psi(t) < \frac{\psi''(t)}{2} (t-1)^2 .\end{aligned}$$

cas 2 : si $t > 1$, alors

$$f''(t) = -(t-1) \psi'''(t) > 0,$$

donc

$$\begin{aligned}f'(t) &= \psi'(t) - (t-1) \psi''(t) > 0, \\ f(t) &= 2\psi(t) - (t-1) \psi'(t) > 0,\end{aligned}$$

donc

$$\begin{aligned}\psi'(t) &> (t-1) \psi''(t), \\ \psi(t) &> \frac{1}{2} (t-1) \psi'(t),\end{aligned}$$

alors $\frac{1}{2} (t-1)^2 \psi''(t) < \psi(t)$, et en utilisant le développement de Taylor au voisinage de $t_0 = 1$, on a :

$$\psi(t) = \frac{\psi''(1)}{2} (t-1)^2 + \frac{\psi'''(\xi)}{6} (t-1)^3, \xi > 1.$$

comme $\psi'''(t) < 0$, donc $\psi(t) < \frac{1}{2} (t-1)^2 \psi''(1)$, alors :

$$\begin{aligned}\frac{\psi'(t)}{2} (t-1) &< \psi(t). \\ \psi''(t) (t-1) &< \psi'(t). \\ \frac{\psi''(t)}{2} (t-1)^2 &< \psi(t) < \frac{\psi''(1)}{2} (t-1)^2 .\end{aligned}$$

2- Suppose que $\psi(t_1) = \psi(t_2)$, tels que $t_1 < 1 < t_2$, et utilisant les cas 1 et cas 2, alors :

$$\frac{\psi''(1)}{2} (t_1 - 1)^2 < \psi(t_1) = \psi(t_2) < \frac{\psi''(1)}{2} (t_2 - 1)^2,$$

comme $\psi''(1) > 0$, alors :

$$1 - t_1 < t_2 - 1,$$

D'autre part, supposons que $-\psi'(t_1) < \psi'(t_2)$, on a :

$$\begin{aligned} \psi(t_2) &> \frac{\psi'(t_2)}{2} (t_2 - 1) \\ &> \frac{\psi'(t_2)}{2} (1 - t_1) \\ &> -\frac{\psi'(t_1)}{2} (1 - t_1) \\ &= \frac{\psi'(t_1)}{2} (t_1 - 1) \\ &> \psi(t_1) \end{aligned}$$

contradiction avec l'hypothèse $\psi(t_2) = \psi(t_1)$, donc $\psi'(t_2) < -\psi'(t_1)$.

Ce qui termine la preuve ■

3.4.5 Borne supérieure de $\Phi(v)$ pour chaque itération externe

Au début de chaque itération externe de l'algorithme, juste avant la mise à jour du paramètre μ avec le facteur $(1 - \theta)$, on a : $\Phi(v) < \tau$. Après la réduction de μ en le multipliant par le facteur $(1 - \theta)$: $\mu^+ = (1 - \theta)\mu$, avec $0 < \theta < 1$, le vecteur v est mis à jour par la relation suivante : $v^+ = \frac{v}{\sqrt{1-\theta}}$. Ceci conduit en général à une augmentation de la valeur de $\Phi(v)$ dans les itérations externes, puis, au cours des itérations internes $\Phi(v)$ diminue jusqu'à ce qu'elle atteigne la première valeur inférieure ou égale à τ . Au cours de l'algorithme, les plus grandes valeurs de $\Phi(v)$ se produisent juste après les mises à jour de μ . Pour cela, nous avons besoin d'une bonne estimation de la borne supérieure de $\Phi(v)$.

Il deviendra clair que dans l'analyse de l'algorithme certaines fonctions inverses en rapport avec les fonctions noyau et leurs dérivées premières jouent un rôle crucial. Nous

introduisons ces fonctions inverses ici.

$\varrho : [0, \infty[\rightarrow [1, +\infty[$ la fonction inverse de ψ .

$\rho : [0, \infty[\rightarrow]0, 1]$ la fonction inverse de $-\frac{1}{2}\psi'$.

Supposons dans toute la suite que ψ est une fonction noyau qualifiée.

Nous avons le résultat suivant.

Théorème 3.4.1 *Pour tout vecteur positif v et tout $\beta > 1$, on a :*

$$\Phi(\beta v) \leq n\psi\left(\beta\varrho\left(\frac{\Phi(v)}{n}\right)\right).$$

Preuve. Soit $\beta > 1$, on définit le problème de maximisation suivant :

$$\max_v \{\Phi(\beta v) : \Phi(v) = z\}$$

tel que $z \geq 0$, Les premières conditions d'optimalité pour ce problème sont :

$$\beta\psi'(\beta v_i) = \lambda\psi'(v_i), \quad i = 1, \dots, n,$$

où λ désigne le multiplicateur de Lagrange.

Comme $\psi'(1) = 0$ et $\beta\psi'(\beta) > 0$, donc $v_i \neq 1$ pour tout $i = 1, \dots, n$. Nous pouvons supposer que $v_i > 1$ pour tout $i = 1, \dots, n$. Si $\psi(v_i) = z_i$ avec $z_i > 0$, cette équation admet deux solutions : $v_i > 1$ et $w_i < 1$. Comme conséquence de lemme **3.4.1** on a $\psi(\beta w_i) < \psi(\beta v_i)$. Puis nous maximisons $\Phi(\beta v)$, il en résulte que nous prenons $v_i > 1$. D'autre part $\lambda > 0$ car $\lambda = \frac{\beta\psi'(\beta v_i)}{\psi'(v_i)}$, $i = 1, \dots, n$. On définit :

$$g(t) = \frac{\psi'(t)}{\psi'(\beta t)}, \quad t > 1,$$

et

$$g'(t) = \frac{\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t)}{(\psi'(\beta t))^2},$$

Or ψ vérifie la condition (3.e), alors $g'(t) > 0$ pour $t > 1$ et $\beta > 1$. Ainsi, nous avons

montré que g est strictement croissante.

Comme

$$g(v_i) = \frac{\beta}{\lambda}, \quad i = 1, \dots, n.$$

alors $v_i = t > 1$, pour $i = 1, \dots, n$. Donc

$$\Phi(v) = z \iff n\psi(t) = z.$$

Alors

$$t = \varrho\left(\frac{z}{n}\right) = \varrho\left(\frac{\Phi(v)}{n}\right).$$

Soit

$$\Phi(\beta v^+) = \max_v \{\Phi(\beta v) : \Phi(v) = z\},$$

on a

$$v^+ = te \text{ et } \Phi(\beta te) = n\psi(\beta t) = n\psi\left(\beta \varrho\left(\frac{z}{n}\right)\right) = n\psi\left(\beta \varrho\left(\frac{\Phi(v)}{n}\right)\right).$$

Ce qui termine la preuve ■

En conséquence de Théorème **3.4.1** on a que si $\Phi(v) \leq \tau$ et $\beta = \frac{1}{\sqrt{1-\theta}}$ alors, après la μ -mise à jour, $\Phi(v)$ est égale à $\Phi\left(\frac{v}{\sqrt{1-\theta}}\right)$ qui est bornée par

$$L_\psi(n, \theta, \tau) = n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right),$$

Comme ϱ croissante, alors

$$\Phi(v) \leq \tau \iff \varrho\left(\frac{\Phi(v)}{n}\right) \leq \varrho\left(\frac{\tau}{n}\right).$$

Corollaire 3.4.1 *Pour tout vecteur positif v , si $\Phi(v) \leq \tau$ et $\frac{1}{\sqrt{1-\theta}} > 1$, on a :*

$$L_\psi(n, \theta, \tau) \leq (\Phi)_0, \\ (\Phi)_0 = \frac{n}{2}\psi''(1) \left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2.$$

Preuve. Comme le vecteur v positif et $\frac{1}{\sqrt{1-\theta}} > 1$, en utilisant le Théorème **3.4.1** pour $\beta = \frac{1}{\sqrt{1-\theta}}$ et $\Phi(v) \leq \tau$ on a

$$L_\psi(n, \theta, \tau) = n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right),$$

Comme ψ vérifie la condition (3.c) alors d'après lemme **3.4.5** cas **2**, on a :

$$L_\psi(n, \theta, \tau) \leq \frac{n}{2}\psi''(1)\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2.$$

Ce qui termine la preuve ■

3.4.6 Analyse de la décroissance de la fonction barrière de proximité Φ

Dans cette sous-section, on va calculer le pas de déplacement et on prouve la décroissance de la fonction barrière de proximité Φ à chaque itération interne et on donne les résultats de complexité de l'algorithme.

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

$$x^+ = x + \alpha\Delta x, \quad s^+ = s + \alpha\Delta s, \quad y^+ = y + \alpha\Delta y.$$

On utilise (3.6), on obtient :

$$x^+ = x\left(e + \alpha\frac{\Delta x}{x}\right) = x\left(e + \alpha\frac{d_x}{v}\right) = \frac{x}{v}(v + \alpha d_x)$$

et

$$s^+ = s\left(e + \alpha\frac{\Delta s}{s}\right) = s\left(e + \alpha\frac{d_s}{v}\right) = \frac{s}{v}(v + \alpha d_s),$$

on a

$$\begin{aligned} v_+ &= \sqrt{\frac{x^+ s^+}{\mu}} \\ &= \sqrt{(v + \alpha d_x)(v + \alpha d_s)}, \end{aligned}$$

pour tout $\alpha > 0$ on pose

$$f(\alpha) = \Phi(v_+) - \Phi(v).$$

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

D'après lemme **3.4.2** (i), on a

$$\begin{aligned} \Phi(v_+) &= \Phi(\sqrt{(v + \alpha d_x)(v + \alpha d_s)}) \\ &\leq \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)), \end{aligned}$$

d'où

$$f(\alpha) \leq f_1(\alpha),$$

tel que :

$$f_1(\alpha) = \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)) - \Phi(v), \quad (3.13)$$

avec

$$f(0) = f_1(0) = 0,$$

la dérivée de f_1 en point α est :

$$f_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi'(v_i + \alpha[d_x]_i)[d_x]_i + \psi'(v_i + \alpha[d_s]_i)[d_s]_i),$$

où $[d_x]_i$ et $[d_s]_i$ désigne respectivement la $i^{\text{ème}}$ composante des vecteurs d_x et d_s .

On utilise (3.12), on obtient :

$$\begin{aligned}
f_1'(0) &= \frac{1}{2} \nabla \Phi(v)^t (d_x + d_s) \\
&= -\frac{1}{2} \nabla \Phi(v)^t \nabla \Phi(v) \\
&= -2(\delta(v))^2.
\end{aligned}$$

La dérivée de f_1' en point α est

$$f_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n (\psi''(v_i + \alpha[d_x]_i)[d_x]_i^2 + \psi''(v_i + \alpha[d_s]_i)[d_s]_i^2), \quad (3.14)$$

d'où $f_1''(\alpha) > 0$, si d_x , ou $d_s \neq 0$, alors dans ce cas f_1 est strictement convexe.

On note par $\delta = \delta(v)$, $\Phi = \Phi(v)$ et $v_{\min} = \min_i(v_i)$.

Lemme 3.4.6 ([3]) Soit $f_1(\alpha)$ défini dans (3.13) et δ défini dans (3.12), alors on a :

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

Preuve. On sait que d_x et d_s vérifient (3.9), par conséquent d_x et d_s sont orthogonaux, implique que

$$\|d_x + d_s\| = \|(d_x, d_s)\| = 2\delta,$$

donc

$$\|d_x\| \leq 2\delta,$$

$$\|d_s\| \leq 2\delta.$$

Alors

$$v_i + \alpha[d_x]_i \geq v_{\min} - 2\alpha\delta, \quad v_i + \alpha[d_s]_i \geq v_{\min} - 2\alpha\delta, \quad i = 1, \dots, n. \quad (3.15)$$

D'après (3.c) (ψ'' strictement décroissante) et (3.14), on obtient :

$$\begin{aligned}
f_1''(\alpha) &\leq \frac{1}{2} \psi''(v_{\min} - 2\alpha\delta) \sum_{i=1}^n [d_x]_i^2 + [d_s]_i^2 \\
&= 2\delta^2 \psi''(v_{\min} - 2\alpha\delta).
\end{aligned}$$

Ce qui termine la preuve ■

Lemme 3.4.7 ([3]) *Si le pas de déplacement α vérifie l'inégalité suivante*

$$-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \leq 2\delta, \quad (3.16)$$

alors :

$$f'_1(\alpha) \leq 0.$$

Preuve. On a

$$\begin{aligned} f'_1(\alpha) &= f'_1(0) + \int_0^\alpha f''_1(\zeta) d\zeta \\ &\leq -2\delta^2 + 2\delta^2 \int_0^\alpha \psi''(v_{\min} - 2\zeta\delta) d\zeta \\ &= -2\delta^2 - \delta \int_0^\alpha \psi''(v_{\min} - 2\zeta\delta) d(v_{\min} - 2\zeta\delta) \\ &= -2\delta^2 + \delta (\psi'(v_{\min}) - \psi'(v_{\min} - 2\alpha\delta)), \end{aligned}$$

et comme

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

alors $f'_1(\alpha) \leq 0$.

Ce qui termine la preuve. ■

Lemme 3.4.8 ([3]), *L'estimation de la plus grande valeur de α vérifiant (3.16) est donnée par*

$$\bar{\alpha} = \frac{1}{2\delta}(\rho(\delta) - \rho(2\delta)).$$

Preuve. Nous voulons déterminer la plus grande valeur de α vérifiant (3.16). On fixe δ , v_{\min} et α .

Comme ψ'' est strictement décroissante alors

$$-\psi''(v_{\min} - 2\alpha\delta) + \psi''(v_{\min}) \leq 0.$$

Donc la valeur maximale pour l'expression à gauche dans (3.16) par rapport à v_{\min} est atteinte si v_{\min} atteint sa valeur minimale,

et d'autre part la dérivée de l'expression à gauche dans (3.16) par rapport à α est

$$2\delta\psi''(v_{\min} - 2\alpha\delta) \geq 0.$$

Donc la valeur maximale pour l'expression à gauche dans (3.16) par rapport à α est atteinte pour la plus grande valeur de α .

Pour $v_{\min} \in]0, 1[$ alors

$$\delta = \frac{1}{2} \|\nabla \Phi(v)\| \geq \frac{1}{2} |\psi'(v_{\min})| \geq -\frac{1}{2} \psi'(v_{\min}),$$

comme ρ est strictement décroissante alors $\rho(\delta) \leq v_{\min}$. Nous prenons

$$v_{\min} = \rho(\delta). \quad (3.17)$$

Donc $\psi'(v_{\min}) = -2\delta$, en le remplaçant dans l'expression à gauche dans (3.16), on obtient

$$-\psi'(v_{\min} - 2\alpha\delta) \leq 4\delta,$$

nous prenons

$$-\frac{1}{2}\psi'(v_{\min} - 2\alpha\delta) = 2\delta,$$

donc

$$v_{\min} - 2\alpha\delta = \rho(2\delta). \quad (3.18)$$

D'après (3.17) et (3.18), on a :

$$v_{\min} = \rho(\delta),$$

$$2\alpha\delta = v_{\min} - \rho(2\delta),$$

donc

$$\alpha = \frac{1}{2\delta}(\rho(\delta) - \rho(2\delta)).$$

Ce qui termine la preuve. ■

Lemme 3.4.9 *Soient ρ et $\bar{\alpha}$ définis dans le lemme 3.4.8 alors on a :*

$$\bar{\alpha} \geq \frac{1}{\psi''(\rho(2\delta))} = \tilde{\alpha}. \quad (3.19)$$

Preuve. Par la définition de ρ on a

$$-\psi'(\rho(\delta)) = 2\delta,$$

on dérive par rapport à δ , on trouve

$$-\psi''(\rho(\delta))\rho'(\delta) = 2,$$

alors

$$\rho'(\delta) = -\frac{2}{\psi''(\rho(\delta))} < 0,$$

et comme la fonction définie par $\psi''(\rho(s))$ est croissante, donc

$$\begin{aligned} \bar{\alpha} &= \frac{1}{2\delta} \int_{2\delta}^{\delta} \rho'(s) ds \\ &= \frac{1}{\delta} \int_{\delta}^{2\delta} \frac{ds}{\psi''(\rho(s))} \\ &\geq \frac{1}{\delta} \frac{1}{\psi''(\rho(2\delta))} \int_{\delta}^{2\delta} ds, \end{aligned}$$

d'où

$$\psi''(\rho(2\delta)) = \max_{[\delta, 2\delta]} \psi''(\rho(s))$$

ce qui donne

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

Ce qui termine la preuve ■

Lemme 3.4.10 ([40]) *On suppose que h est une fonction convexe et deux fois différentiable avec*

$$h(0) = 0, \quad h'(0) < 0,$$

et h atteint son minimum global à $t^ > 0$, et h'' est croissante pour tout $t \in [0, t^*]$, alors pour tout $t \in [0, t^*]$, on a*

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

Preuve. On suppose que h'' est croissante pour tout $t \in [0, t^*]$, Posons $\gamma = h'$, alors γ' est croissante, c'est à dire

$$[\gamma'(t_2) - \gamma'(t_1)][t_2 - t_1] \geq 0, \quad \forall t_1, t_2 \in [0, t^*]$$

L'opérateur γ' est donc monotone. Se rappeler que la condition

$$\langle \nabla f(x_2) - \nabla f(x_1), x_2 - x_1 \rangle \geq 0$$

est une **CNS** de convexité de la fonction f .

Donc $\gamma = h'$ est une fonction convexe. La sécante passant par les points $(0, h'(0))$ et $(t^*, h'(t^*))$ est au dessus de la courbe $(t, h'(t))$ pour tout $t \in [0, t^*]$. L'équation de la sécante passant par les deux points est

$$sec(t) = h'(0) + t \frac{h'(t^*) - h'(0)}{t^*}.$$

Se rappeler que $h'(t^*) = 0$. On a donc

$$h'(t) \leq sec(t) = \left(1 - \frac{t}{t^*}\right) h'(0), \quad \forall t \in [0, t^*].$$

Alors

$$\int_0^t h'(\zeta) d\zeta \leq \int_0^t \left(1 - \frac{\zeta}{t^*}\right) h'(0) d\zeta, \quad \forall t \in [0, t^*].$$

donc

$$[h(t) - h(0)] \leq \frac{th'(0)}{2} + \frac{th'(0)}{2} \left(1 - \frac{t}{t^*}\right), \quad \forall t \in [0, t^*].$$

Se rappeler que $h(0) = 0$. On a donc

$$h(t) - \frac{th'(0)}{2} \leq \frac{th'(0)}{2} \left(1 - \frac{t}{t^*}\right), \quad \forall t \in [0, t^*].$$

Comme $h'(0) < 0$, alors

$$h(t) - \frac{th'(0)}{2} \leq 0, \quad \forall t \in [0, t^*].$$

Ce qui termine la preuve ■

Lemme 3.4.11 *Si le pas de déplacement α vérifie $\alpha \leq \bar{\alpha}$ alors*

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

Preuve. Soit h définie par

$$h(\alpha) = -2\alpha\delta^2 + \alpha\delta\psi'(v_{\min}) - \frac{1}{2}\psi(v_{\min}) + \frac{1}{2}\psi(v_{\min} - 2\alpha\delta),$$

alors

$$\begin{aligned} h(0) &= f_1(0) = 0, \quad h'(0) = f'_1(0) = -2\delta^2 \\ h''(\alpha) &= 2\delta^2\psi''(v_{\min} - 2\alpha\delta) \end{aligned}$$

d'après le lemme **3.4.6** on a

$$f''_1(\alpha) \leq h''(\alpha)$$

et par conséquent on obtient

$$f_1'(\alpha) \leq h'(\alpha) \text{ et } f_1(\alpha) \leq h(\alpha),$$

donc si $\alpha \leq \bar{\alpha}$ on a

$$\begin{aligned} h'(\alpha) &= h'(0) + \int_0^\alpha h''(\xi) d\xi, \\ &= -2\delta^2 + \int_0^\alpha 2\delta^2 \psi''(v_{\min} - 2\xi\delta) d\xi \\ &= -2\delta^2 + \delta(-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min})) \leq -2\delta^2 + 2\delta^2 = 0, \end{aligned}$$

avec h'' est croissante, donc d'après le lemme **3.4.10** on a

$$f(\alpha) \leq f_1(\alpha) \leq h(\alpha) \leq \frac{\alpha h'(0)}{2} = -\alpha\delta^2.$$

Ce qui termine la preuve. ■

En combinant les résultats des lemmes **3.4.9** et **3.4.11** on obtient

$$f(\tilde{\alpha}) \leq -\frac{\delta^2}{\psi''(\rho(2\delta))}. \quad (3.20)$$

3.4.7 Analyse de la complexité algorithmique de l'algorithme

Nous avons besoin des lemmes ci-dessous, utilisés par J. Peng et al. [40], pour faciliter au lecteur la compréhension des propos établis le long des différentes preuves.

Lemme 3.4.12 ([40]) *Si $\alpha \in [0, 1]$ alors*

$$(1+t)^\alpha \leq 1 + \alpha t, \quad \forall t \geq -1.$$

Preuve. On définit la fonction

$$f(t) = (1+t)^\alpha - 1 - \alpha t, \text{ pour } t \geq -1.$$

Alors

$$\begin{aligned} f'(t) &= \alpha(1+t)^{\alpha-1} - \alpha. \\ f''(t) &= \alpha(\alpha-1)(1+t)^{\alpha-2}. \\ f''(t) &\leq 0, \end{aligned}$$

donc f est concave et $f'(0) = 0$, alors

$$f(t) \leq f(0) = 0.$$

Ce qui termine la preuve. ■

Lemme 3.4.13 ([40]). 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 :

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

Preuve. En utilisant de lemme 3.4.12, nous pouvons écrire

$$\begin{aligned} t_{k+1}^\gamma &\leq (t_k - \beta t_k^{1-\gamma})^\gamma \\ &= t_k^\gamma (1 - \beta t_k^{-\gamma})^\gamma \\ &\leq t_k^\gamma (1 - \gamma \beta t_k^{-\gamma}) \\ &= t_k^\gamma - \gamma \beta, \end{aligned}$$

donc $t_k^\gamma \leq t_0^\gamma - \gamma \beta k$, en remplaçant k par K , on obtient $t_0^\gamma - \gamma \beta K > 0$, alors

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

Ce qui termine la preuve. ■

On note par $(\Phi)_0$ la première μ -mise à jour de $\Phi(v)$ et Φ_k , $k = 1, 2, \dots, K$ suite des valeurs de $\Phi(v)$ dans les itérations internes. Alors d'après le corollaire **3.4.1** on a :

$$(\Phi)_0 = \frac{n}{2} \psi''(1) \left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1 \right)^2.$$

D'autre part, d'après (3.13) et (3.20) on a

$$\Phi_{k+1} - \Phi_k \leq f(\tilde{\alpha}) \leq -\frac{\delta^2}{\psi''(\rho(2\delta))}. \quad (3.21)$$

alors on suppose qu'ils existent $\beta > 0$ et $0 < \gamma \leq 1$, tels que

$$-\frac{\delta^2}{\psi''(\rho(2\delta))} \leq -\beta \Phi_k^{1-\gamma} \leq -\beta (\Phi_k - \tau)^{1-\gamma}.$$

Alors

$$(\Phi_{k+1} - \tau) - (\Phi_k - \tau) \leq -\beta (\Phi_k - \tau)^{1-\gamma},$$

donc

$$(\Phi_{k+1} - \tau) \leq (\Phi_k - \tau) - \beta (\Phi_k - \tau)^{1-\gamma},$$

en utilisant le lemme **3.4.13** pour $t_k = \Phi_k - \tau > 0$, on a K le nombre des itérations internes de chaque itération externe, défini comme suit :

$$K \leq \frac{[(\Phi)_0 - \tau]^\gamma}{\beta\gamma} \leq \frac{[(\Phi)_0]^\gamma}{\beta\gamma} = \frac{\left(\frac{n}{2} \psi''(1) \left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1 \right)^2 \right)^\gamma}{\beta\gamma}. \quad (3.22)$$

La détermination de nombre total d'itérations nécessaires pour trouver une solution optimale primale-duale nécessite le calcul du nombre d'itérations externes.

Théorème 3.4.2 *Soit k le nombre d'itérations internes nécessaires pour trouver une*

solution optimale primale-duale approchée à une précision $\epsilon > 0$, alors on a :

$$k \geq \frac{1}{\theta} \log \frac{n}{\epsilon}. \quad (3.23)$$

Preuve. Si le paramètre de la trajectoire centrale avec $\mu^0 = 1$, et $\mu_k = (1 - \theta)^k \mu^0$, $n\mu \leq \epsilon$, on a :

$$\begin{aligned} (1 - \theta)^k \mu^0 n \leq \epsilon &\Rightarrow (1 - \theta)^k \leq \frac{\epsilon}{n} \\ &\Rightarrow k \log(1 - \theta) \leq \log \frac{\epsilon}{n} \end{aligned}$$

Puisque $-\log(1 - \theta) \geq \theta$ on obtient

$$k\theta \geq \log \frac{n}{\epsilon}$$

$$k \geq \frac{1}{\theta} \log \frac{n}{\epsilon}.$$

Ce qui termine la preuve. ■

Le nombre total d'itérations nécessaires pour trouver une solution optimale primale-duale approchée avec des précisions $\epsilon > 0$ et $\tau > 0$ n'est autre que la multiplication de nombre d'itérations externes par le nombre d'itérations internes. Nous obtenons le nombre total d'itérations par l'équation suivante :

$$\frac{[(\Phi)_0]^\gamma}{\beta\gamma\theta} \log \frac{n}{\epsilon} = \frac{\left(\frac{n}{2} \psi''(1) \left(\frac{\varrho(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1 \right)^2 \right)^\gamma}{\beta\gamma\theta}. \quad (3.24)$$

Si on prend $\tau = \mathbf{O}(n)$ et $\theta = \Theta(1)$, dans (3.24), on trouve le nombre d'itérations pour **IPMs** à long-pas.

Par contre, si $\tau = \mathbf{O}(1)$ et $\theta = \Theta(\frac{1}{\sqrt{n}})$, dans (3.24), on trouve le nombre d'itérations pour **IPMs** à court-pas.

3.5 Conclusion

Dans ce chapitre, nous avons apporté des aménagements d'ordre théorique et algorithmique sur la méthode de trajectoire centrale via une fonction noyau. Cette approche a l'avantage de démarrer avec n'importe quel point (x^0, y^0, s^0) satisfaisant la condition **IPC**. Par contre, elle a un inconvénient représenté par sa complexité algorithmique. Cette complexité est égale à celle de la méthode de trajectoire centrale classique (c'est-à-dire le nombre d'itérations externes) multiplié par le nombre d'itérations internes. Il serait probablement plus intéressant d'insister plus sur la forme analytique de la fonction noyau, pour réduire la complexité algorithmique (c'est-à-dire réduire le nombre d'itérations internes). C'est le but de notre prochain chapitre.

En effet, dans le chapitre suivant, nous nous intéressons à la direction de déplacement, c'est-à-dire la forme analytique de la fonction noyau, dans l'algorithme de trajectoire centrale via une fonction noyau pour résoudre un programme linéaire. Nous proposons deux nouvelles fonctions noyau avec paramètres pour améliorer la complexité algorithmique. Ces nouveaux résultats sont d'ordre théorique, algorithmique et numérique.

Chapitre 4

Famille des nouvelles fonctions noyau et algorithmes (TC) corespondant

4.1 Introduction

Plusieurs travaux montrent que l'itération théorique liée court-pas dans les **IPMs** est meilleure que celle liée long-pas. Cependant, il est devenu clair que la théorie de la complexité algorithmique des méthodes à long-pas, qui sont les méthodes les plus efficaces dans la pratique, peut être nettement améliorée en utilisant d'autres fonctions barrière.

La plupart des algorithmes **IPMs** pour l'optimisation linéaire sont basés sur la fonction barrière logarithmique [20, 45]. En 2001, Peng et al. dans [39] ont proposé des nouvelles variantes de **IPMs** basées sur une nouvelle fonction noyau non logarithmique. Une telle fonction est fortement convexe et coercive; lisse sur son domaine qui n'est autre que l'ortan positif. Ils ont obtenu les résultats de la complexité algorithmique : $\mathbf{O}\left(qn^{\frac{q+1}{2q}} \log \frac{n}{\epsilon}\right)$ est la complexité des méthodes à long-pas et $\mathbf{O}\left(q^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$ est la complexité des méthodes à court-pas. Pour $q = \frac{\log n}{2}$, on obtient la meilleure borne pour les méthodes à long-pas, soit $\mathbf{O}\left(\sqrt{n} (\log n) \log \frac{n}{\epsilon}\right)$. Cette borne est la meilleure amélioration

obtenue jusqu'à présent.

En 2002, Y.Q. Bai et al. dans [5] ont introduit une nouvelle fonction noyau non logarithmique. Ils ont obtenu une complexité de $\mathbf{O}\left(qn \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à long-pas et $\mathbf{O}\left(q^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$ pour les méthodes à court-pas. En 2004, Y.Q. Bai et al. dans [3] ont présenté des nouvelles fonctions noyau non logarithmiques. Ils ont obtenu $\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$ itérations comme borne pour les méthodes à long-pas et $\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à court-pas. Dans le même article, Bai et al. ont introduit pour la première fois une nouvelle fonction noyau avec un terme barrière trigonométrique.

En 2012, El Ghami et al. dans [19] ont évalué la fonction noyau avec un terme barrière trigonométrique introduite précédemment par Bai et al. dans [3]. Ils ont obtenu une borne de $\mathbf{O}\left(n^{\frac{3}{4}} \log \frac{n}{\epsilon}\right)$ itérations pour les méthodes à long-pas. Cette découverte a encouragé les chercheurs à améliorer cette borne pour les fonctions noyau avec un terme barrière trigonométrique (voir [19, 3, 42, 41, 30, 33]). La meilleure amélioration a été donnée par Bouafia et al. dans (voir [11]).

L'idée principale dans **IPMs** est de trouver une nouvelle direction dans la méthode de trajectoire centrale via une fonction noyau basée sur le remplacement de la fonction barrière Φ dans (3.10) par la nouvelle fonction barrière Φ_{New} . Par conséquence, la fonction noyau ψ , définie en (3.11), est remplacée par la nouvelle fonction noyau ψ_{New} comme suit :

$$d_x + d_s = -\nabla \Phi_{New}(v), \text{ où } \Phi_{New}(v) = \sum_{i=1}^n \psi_{New}(v_i).$$

Par conséquence, la nouvelle direction de recherche $(\Delta x, \Delta y, \Delta s)$ est obtenue par le système de Newton modifié suivant :

$$\begin{aligned} \overline{A}d_x &= 0, \\ \overline{A}^t \Delta y + d_s &= 0, \\ d_x + d_s &= -\nabla \Phi_{New}(v). \end{aligned} \tag{4.*}$$

On a d_x et d_s sont orthogonaux parce que le vecteur d_x appartient au noyau de la matrice \overline{A} et le vecteur d_s appartient à son image.

Comme d_x et d_s sont orthogonaux, on a

$$\begin{aligned} d_x = d_s = 0 &\iff \nabla \Phi_{New}(v) = 0 \\ &\iff v = e \\ &\iff \Phi_{New}(v) = 0 \\ &\iff x = x(\mu), s = s(\mu). \end{aligned}$$

Nous utilisons Φ_{New} pour mesurer la distance entre l'itéré μ -centre et la trajectoire centrale pour $\mu > 0$ donné. Nous définissons également la mesure de proximité basée sur la norme, $\delta : \mathbb{R}_{++}^n \rightarrow \mathbb{R}_+$, comme suit :

$$\delta(v) = \frac{1}{2} \|\nabla \Phi_{New}(v)\| = \frac{1}{2} \|d_x + d_s\|. \quad (4.**)$$

On donne dans le tableau suivant les différentes fonctions noyau connues et les deux nouvelles fonctions noyau, qu'on a proposées (ψ_M, ψ_B), et la complexité algorithmique de chacune.

Afin d'établir la complexité des algorithmes primaux-duaux basés sur nos deux nouvelles fonctions noyau, et pour la démonstration de tous les résultats de convergence, nous allons suivre dans toute la suite les mêmes démarches acheminées par Peng et Bai (voir [39, 3]).

i	Fonction noyau ψ_i	Long-pas	Court-pas	Ref
1	$\frac{t^2-1}{2} - \log t.$	$\mathbf{O}\left(n \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[45]
2	$\frac{t^{1+p}-1}{1+p} - \log t, \quad p \in [0, 1].$	$\mathbf{O}\left(n \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[20]
3	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}, \quad q > 1.$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(q^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$	[39]
4	$\frac{t^2-1}{2} + e^{\frac{1}{t}-1} - 1.$	$\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[3]
5	$t - 1 + \frac{t^{1-q}-1}{q-1}, \quad q > 1.$	$\mathbf{O}\left(qn \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(q^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$	[5]
6	$\frac{t^2-1}{2} + \frac{6}{\pi} \tan\left(\pi \frac{1-t}{4t+2}\right).$?	?	[3]
7	$\frac{t^2-1}{2} + \frac{6}{\pi} \tan\left(\pi \frac{1-t}{4t+2}\right).$	$\mathbf{O}\left(n^{\frac{3}{4}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[19]
8	$\frac{t^2-1}{2} - \log t + \frac{1}{8} \tan^2\left(\pi \frac{1-t}{4t+2}\right).$	$\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$	— — —	[42]
9	$\frac{t^2-1}{2} - \int_1^t e^{3\left(\left(\tan \frac{\pi}{2x+2}\right)-1\right)} dx.$	$\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$	— — —	[41]
10	$\frac{t^2-1}{2} - \log t + \lambda \tan^2 h(t),$ $h(t) = \frac{\pi(1-t)}{(3t+2)}, 0 < \lambda \leq \frac{8}{25}.$	$\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$	— — —	[13]
11	$t^2 - 2t + \frac{1}{\sin h(t)},$ $h(t) = \frac{\pi t}{1+t}.$	$\mathbf{O}\left(n^{\frac{3}{4}} \log \frac{n}{\epsilon}\right)$	— — —	[30]
12	$\frac{(t-1)^2}{2} + \frac{(t-1)^2}{2t} + \frac{1}{8} \tan^2 h(t),$ $h(t) = \frac{\pi(1-t)}{(2+4t)}.$	$\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$	— — —	[33]
M	$\frac{p}{2}t^2 + e^{p(\frac{1}{t}-1)} - (1 + \frac{p}{2}), \quad p > 0.$	$\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[10]
B	$\frac{t^2-1}{2} + \frac{4}{\pi p} \left[\tan^p\left(\frac{\pi}{2t+2}\right) - 1 \right], \quad p \geq 2.$	$\mathbf{O}\left(pn^{\frac{p+2}{2(p+1)}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(p^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$	[11]

Tableau 1 : Résultats de la complexité algorithmique pour l'optimisation linéaire.

4.2 Fonctions noyau avec un terme barrière exponentiel

4.2.1 Propriétés de fonction noyau

Dans cette partie, nous présentons une nouvelle fonction noyau paramétrée et nous donnons ses propriétés qui sont essentielles à notre analyse de la complexité.

Nous rappelons que notre fonction paramétrée indiquée dans le tableau précédent est ψ_M , définie par :

$$\psi_M(t) = p\left(\frac{t^2 - 1}{2}\right) + e^{p(\frac{1}{t}-1)} - 1, \quad (4.1)$$

où le paramètre p est un nombre réel positif. Nous donnons les trois premières dérivées par rapport à t :

$$\begin{aligned} \psi'_M(t) &= pt - \frac{p}{t^2} e^{p(\frac{1}{t}-1)}, \\ \psi''_M(t) &= p + \left(\frac{2p}{t^3} + \frac{p^2}{t^4}\right) e^{p(\frac{1}{t}-1)}, \\ \psi'''_M(t) &= -\left(\frac{6p}{t^4} + \frac{6p^2}{t^5} + \frac{p^3}{t^6}\right) e^{p(\frac{1}{t}-1)}. \end{aligned} \quad (4.2)$$

De toute évidence, ψ_M est une fonction noyau et

$$\psi''_M(t) > p. \quad (4.3)$$

Lemme 4.2.1 *pour ψ_M , nous avons les résultats suivants :*

(i) ψ_M est exponentielle convexe pour tout $t > 0$, et

$$\psi_M(\sqrt{t_1 t_2}) \leq \frac{1}{2} (\psi_M(t_1) + \psi_M(t_2)).$$

(ii) ψ''_M est monotone décroissante pour tout $t > 0$.

(iii) $t\psi''_M(t) - \psi'_M(t) > 0$ pour tout $t > 0$.

(iv) $\psi''_M(t)\psi'_M(\beta t) - \beta\psi'_M(t)\psi''_M(\beta t) > 0$, $t > 1$, $\beta > 1$.

Preuve. Pour (i), en utilisant (4.2), on a

$$t\psi_M''(t) + \psi_M'(t) = 2pt + \left(\frac{p}{t^2} + \frac{p^2}{t^3}\right) e^{p(\frac{1}{t}-1)} > 0 \text{ pour tout } t > 0,$$

et par le lemme **3.4.2**, on trouve le résultat.

Pour (ii), en utilisant (4.2), on a $\psi_M'''(t) < 0$, nous avons donc le résultat.

Pour (iii), en utilisant (4.2), on a

$$t\psi_M''(t) - \psi_M'(t) = \left(\frac{3p}{t^2} + \frac{p^2}{t^3}\right) e^{p(\frac{1}{t}-1)} > 0 \text{ pour tout } t > 0.$$

Pour (iv), en utilisant le lemme **3.4.4**, (ii) et (iii), on obtient le résultat.

Ce qui termine la preuve. ■

Lemme 4.2.2 Pour ψ_M , on a

$$\frac{p}{2}(t-1)^2 \leq \psi_M(t) \leq \frac{1}{2p} [\psi_M'(t)]^2, \quad t > 0 \quad (4.4)$$

$$\psi_M(t) \leq \frac{p^2 + 3p}{2}(t-1)^2, \quad t > 1 \quad (4.5)$$

Preuve. Pour (4.4), en utilisant (4.3), on a

$$\psi_M(t) = \int_1^t \int_1^x \psi_M''(y) dy dx \geq \int_1^t \int_1^x p dy dx = \frac{p}{2}(t-1)^2$$

$$\begin{aligned} \psi_M(t) &= \int_1^t \int_1^x \psi_M''(y) dy dx \\ &\leq \frac{1}{p} \int_1^t \int_1^x \psi_M''(y) \psi_M''(x) dy dx \\ &= \frac{1}{p} \int_1^t \psi_M''(x) \psi_M'(x) dx \\ &= \frac{1}{p} \int_1^t \psi_M'(x) d\psi_M'(x) \\ &= \frac{1}{2p} [\psi_M'(t)]^2 \end{aligned}$$

Pour (4.5), comme $\psi_M(1) = \psi'_M(1) = 0$, $\psi'''_M(t) < 0$, $\psi''_M(1) = p^2 + 3p$, et en utilisant le théorème de Taylor, nous avons

$$\begin{aligned}\psi_M(t) &= \psi_M(1) + \psi'_M(1)(t-1) + \frac{1}{2}\psi''_M(1)(t-1)^2 + \frac{1}{6}\psi'''_M(\xi)(\xi-1)^3 \\ &= \frac{1}{2}\psi''_M(1)(t-1)^2 + \frac{1}{6}\psi'''_M(\xi)(\xi-1)^3 \\ &\leq \frac{1}{2}\psi''_M(1)(t-1)^2 \\ &= \frac{p^2+3p}{2}(t-1)^2\end{aligned}$$

pour certains ξ vérifiant : $1 \leq \xi \leq t$.

Ce qui termine la preuve. ■

Soit $\varrho : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de ψ_M pour $t \geq 1$ et soit $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de $\frac{-1}{2}\psi'_M$ pour tout $t \in]0, 1]$. Alors nous avons le lemme suivant :

Lemme 4.2.3 *Pour ψ_M , on a*

$$1 + \sqrt{\frac{2}{p^2 + 3p}}s \leq \varrho(s) \leq 1 + \sqrt{\frac{2}{p}}s, \quad s \geq 0 \quad (4.6)$$

$$\rho(z) > \frac{1}{\sqrt{\frac{2}{p}z + 1}}, \quad z \geq 0 \quad (4.7)$$

Preuve. Pour montrer (4.6), soit $s = \psi_M(t)$, $t \geq 1$, i.e., $\varrho(s) = t$, $t \geq 1$. Par la définition de ψ_M , on a

$$s = \frac{p}{2}t^2 + e^{p(\frac{1}{t}-1)} - (1 + \frac{p}{2}), \quad p > 0.$$

En utilisant (4.3) et en prenant $t \geq 1$, on a

$$\begin{aligned}\psi''_M(t) > p &\Leftrightarrow \int_1^t \int_1^x \psi''_M(y) dy dx > \int_1^t \int_1^x p dy dx \\ &\Leftrightarrow \psi_M(t) > \frac{p}{2}(t-1)^2 \\ &\Rightarrow s > \frac{p}{2}(t-1)^2\end{aligned}$$

ce qui implique que

$$t = \varrho(s) \leq 1 + \sqrt{\frac{2}{p}s}.$$

En utilisant (4.5), on a $s = \psi_M(t) \leq \frac{p^2+3p}{2}(t-1)^2$, donc

$$t = \varrho(s) \geq 1 + \sqrt{\frac{2}{p^2+3p}s}.$$

Pour montrer (4.7), soit $z = -\frac{1}{2}\psi'_M(t)$, $t \in]0, 1]$. Par la définition de $\rho : \rho(z) = t$, $t \in]0, 1]$, on a $e^{p(\frac{1}{t}-1)} > 1$. Par la définition de ψ_M , on a

$$\begin{aligned} z &= -\frac{1}{2} \left(pt - \frac{p}{t^2} e^{p(\frac{1}{t}-1)} \right) \\ &> -\frac{1}{2} \left(pt - \frac{p}{t^2} \right) = \frac{p}{2} \left(\frac{1}{t^2} - t \right) \\ &= \frac{p}{2} \left(\left(\frac{1}{t^2} - 1 \right) + (1 - t) \right) \\ &\geq \frac{p}{2} \left(\frac{1}{t^2} - 1 \right) \end{aligned}$$

ce qui implique que

$$t = \rho(z) > \frac{1}{\sqrt{\frac{2}{p}z + 1}}.$$

Ce qui termine la preuve. ■

Soit $\psi_b(t) = \frac{p}{t^2} e^{p(\frac{1}{t}-1)}$, $p > 0$, pour $t \in]0, 1]$ et $\underline{\rho} : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de ψ_b . Alors nous avons le lemme suivant :

Lemme 4.2.4 *Pour ψ_b , on a*

$$\underline{\rho}(z) > \frac{1}{1 + \log \left(\frac{z}{p} \right)^{\frac{1}{p}}}, \quad z \geq 0 \quad (4.8)$$

$$\rho(z) \geq \underline{\rho}(p + 2z), \quad z \geq 0 \quad (4.9)$$

Preuve. Pour montrer (4.8), soit $z = \psi_b(t) = \frac{p}{t^2} e^{p(\frac{1}{t}-1)}$, $p > 0$, pour tout $t \in]0, 1]$.

On a

$$e^{p(\frac{1}{t}-1)} = \frac{t^2 z}{p} \leq \frac{z}{p},$$

ce qui implique que

$$t = \underline{\rho}(z) \geq \frac{1}{1 + \log \left(\frac{z}{p} \right)^{\frac{1}{p}}}, \quad z \geq 0.$$

Pour montrer (4.9), on a

$$z = \frac{-1}{2} \psi'_M(t) = \frac{-1}{2} (pt - \psi_b(t)), \quad z \geq 0,$$

ce qui implique que

$$\psi_b(t) = pt + 2z$$

comme $t \leq 1$, on a

$$\psi_b(t) \leq p + 2z$$

et comme $\underline{\rho}$ est décroissante, on obtient

$$\underline{\rho}(z) = t \geq \underline{\rho}(p + 2z),$$

D'autre part $\rho(z) = t$, alors

$$\rho(z) \geq \underline{\rho}(p + 2z),$$

Ce qui termine la preuve. ■

Lemme 4.2.5 *Soit $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$, Si $\Phi_M(v) \leq \tau$ alors, on a*

$$\Phi_M(v_+) \leq \frac{(p^2 + 3p)}{2(1-\theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}\tau} \right)^2.$$

Preuve. Comme $\frac{1}{\sqrt{1-\theta}} \geq 1$ et $\varrho\left(\frac{\Phi_M(v)}{n}\right) \geq 1$, nous obtenons $\frac{\varrho\left(\frac{\Phi_M(v)}{n}\right)}{\sqrt{1-\theta}} \geq 1$.

En utilisant (4.5), c'est-à-dire

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

le Théorème **3.4.1**, avec $\beta = \frac{1}{\sqrt{1-\theta}}$, l'inégalité (4.6) et $\Phi_M(v) \leq \tau$, on obtient

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

Nous avons utilisé dans l'inégalité précédente, la propriété suivante :

$$1 - \sqrt{1-\theta} = \frac{\theta}{1 + \sqrt{1-\theta}} \leq \theta.$$

Ce qui termine la preuve. ■

Soit

$$(\Phi_M)_0 = \frac{(p^2+3p)}{2(1-\theta)}\left(\theta\sqrt{n}+\sqrt{\frac{2}{p}\tau}\right)^2 = L(n, \theta, \tau), \quad (4.10)$$

alors, $(\Phi_M)_0$ est un majorant de $\Phi_M(v_+)$ dans l'algorithme.

Lemme 4.2.6 *Soit $\delta(v)$ défini dans (3.12). Alors, on a*

$$\delta(v) \geq \sqrt{\frac{p}{2}\Phi_M(v)} = \sqrt{\frac{p}{2}\Phi_M}.$$

Preuve. En utilisant (4.4), on a

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

donc

$$\delta(v) \geq \sqrt{\frac{p}{2}\Phi_M(v)} = \sqrt{\frac{p}{2}\Phi_M}.$$

Ce qui termine la preuve. ■

4.2.2 Analyse de la complexité

Tout au long de cette sous-section, nous supposons que $\tau \geq 1$. En utilisant le lemme 4.2.6 et l'hypothèse que $\Phi_M(v) \geq \tau$, on a

$$\delta(v) \geq \sqrt{\frac{p}{2}}.$$

D'après les lemmes 3.4.6-3.4.8 , on a

Lemme 4.2.7 *Soit $\bar{\alpha}$ celle définie dans le lemme 3.4.8. Si*

$$\Phi_M = \Phi_M(v) \geq \tau \geq 1,$$

alors on a

$$\bar{\alpha} \geq \frac{1}{p + [(2+p)(p+4)\sqrt{\frac{p}{2}\Phi_M}] \left[1 + \log \left(1 + \frac{4}{p} \sqrt{\frac{p}{2}\Phi_M} \right)^{\frac{1}{p}} \right]^2}$$

Preuve. En utilisant le lemme 3.4.9 et la définition de $\psi_M''(t)$ et (4.9), on a

$$\bar{\alpha} \geq \frac{1}{\psi_M''(\rho(2\delta))} \geq \frac{1}{\psi_M''(\rho(p+2(2\delta)))}$$

D'après le lemme 4.2.6, on a $\delta \geq \sqrt{\frac{p}{2}\Phi_M}$, et des fonctions croissantes $\frac{1}{\psi_M''(t)}$ et $\rho(\delta)$, on a :

$$\begin{aligned} \delta \geq \sqrt{\frac{p}{2}\Phi_M} &\Leftrightarrow \rho(2\delta) \geq \rho(2\sqrt{\frac{p}{2}\Phi_M}) \\ &\Leftrightarrow \psi_M''(\rho(2\delta)) \leq \psi_M''(\rho(2\sqrt{\frac{p}{2}\Phi_M})) \\ &\Leftrightarrow \frac{1}{\psi_M''(\rho(2\delta))} \geq \frac{1}{\psi_M''(\rho(2\sqrt{\frac{p}{2}\Phi_M}))} \end{aligned}$$

Alors

$$\begin{aligned}\bar{\alpha} &\geq \frac{1}{\psi''_M(\rho(2\delta))} \\ &\geq \frac{1}{\psi''_M(\rho(2\sqrt{\frac{p}{2}\Phi_M}))} \\ &\geq \frac{1}{\psi''_M(\rho(p+2(2\sqrt{\frac{p}{2}\Phi_M})))}.\end{aligned}$$

Posons $t = \rho(p+2(2\sqrt{\frac{p}{2}\Phi_M})) = \rho(p+4\sqrt{\frac{p}{2}\Phi_M})$ alors nous obtenons $t \leq 1$ et

$$\begin{aligned}\bar{\alpha} &\geq \frac{1}{\psi''_M(t)} \\ &= \frac{1}{p+\left(\frac{2p}{t^3}+\frac{p^2}{t^4}\right)e^{p(\frac{1}{t}-1)}} \\ &= \frac{1}{p+\left(\frac{2t+p}{t^2}\right)\psi_b(t)} \\ &\geq \frac{1}{p+\left(\frac{2+p}{t^2}\right)\psi_b(t)} \geq \frac{1}{p+\left(\frac{2+p}{t^2}\right)\psi_b(t)} \geq \frac{1}{p+(2+p)\frac{1}{t^2}\psi_b(t)}\end{aligned}$$

On a aussi

$$\begin{aligned}\frac{1}{t^2} &= \frac{1}{\left[\rho(p+4\sqrt{\frac{p}{2}\Phi_M})\right]^2} \\ &\leq \left[1 + \log\left(\frac{p+4\sqrt{\frac{p}{2}\Phi_M}}{p}\right)^{\frac{1}{p}}\right]^2 \\ &= \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}\Phi_M}\right)^{\frac{1}{p}}\right]^2\end{aligned}$$

et

$$\psi_b(t) = p + 4\sqrt{\frac{p}{2}\Phi_M} = p\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}\Phi_M}\right).$$

Enfin, nous obtenons

$$\begin{aligned}\bar{\alpha} &\geq \frac{1}{p+(2+p)(p+4\sqrt{\frac{p}{2}\Phi_M})\left[1+\log\left(1+\frac{4}{p}\sqrt{\frac{p}{2}\Phi_M}\right)^{\frac{1}{p}}\right]^2} \\ &\geq \frac{1}{p+\left[(2+p)(p+4)\sqrt{\frac{p}{2}\Phi_M}\right]\left[1+\log\left(1+\frac{4}{p}\sqrt{\frac{p}{2}\Phi_M}\right)^{\frac{1}{p}}\right]^2}.\end{aligned}$$

Ce qui termine la preuve. ■

Soit

$$\tilde{\alpha} = \frac{1}{p + \left[(2+p)(p+4)\sqrt{\frac{p}{2}\Phi_M}\right]\left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}\Phi_M}\right)^{\frac{1}{p}}\right]^2}. \quad (4.11)$$

$\tilde{\alpha}$ est le pas de déplacement et $\tilde{\alpha} \leq \bar{\alpha}$.

Lemme 4.2.8 Pour $\tilde{\alpha}$ le pas de déplacement, défini dans (4.11), et en prenant

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

Alors

$$f\left(\tilde{\alpha}\right) \leq \frac{-\sqrt{\frac{p}{2}} [(\Phi_M)]^{\frac{1}{2}}}{\left[\sqrt{\frac{2}{p}}p + (2+p)(p+4)\right] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}(\Phi_M)_0\right)^{\frac{1}{p}}\right]^2} \quad (4.12)$$

Preuve. En utilisant le lemme 3.4.11 pour $\alpha = \tilde{\alpha}$ et (4.11), on a

$$\begin{aligned} f\left(\tilde{\alpha}\right) &\leq -\tilde{\alpha}\delta^2 \\ &= -\frac{\delta^2}{p + [(2+p)(p+4)\sqrt{\frac{p}{2}}\Phi_M] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}\Phi_M\right)^{\frac{1}{p}}\right]^2} \end{aligned}$$

En utilisant le lemme 4.2.6, on a $\delta \geq \sqrt{\frac{p}{2}\Phi_M}$, et $\sqrt{\frac{p}{2}\Phi_M} \geq \sqrt{\frac{p}{2}}$ donc

$$\begin{aligned} f\left(\tilde{\alpha}\right) &\leq -\frac{\left(\sqrt{\frac{p}{2}\Phi_M}\right)^2}{\sqrt{\frac{p}{2}\Phi_M} \left[\sqrt{\frac{2}{p}}p + (2+p)(p+4)\right] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}\Phi_M\right)^{\frac{1}{p}}\right]^2} \\ &= -\frac{\sqrt{\frac{p}{2}\Phi_M}}{\left[\sqrt{\frac{2}{p}}p + (2+p)(p+4)\right] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}\Phi_M\right)^{\frac{1}{p}}\right]^2} \\ &\leq -\frac{\sqrt{\frac{p}{2}(\Phi_M)}}{\left[\sqrt{\frac{2}{p}}p + (2+p)(p+4)\right] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}(\Phi_M)_0\right)^{\frac{1}{p}}\right]^2}. \end{aligned}$$

Ce qui termine la preuve. ■

Après la mise à jour de μ à $(1 - \theta)\mu$, on a

$$\Phi_M(v_+) \leq \frac{(p^2 + 3p)}{2(1 - \theta)} \left(\theta\sqrt{n} + \sqrt{\frac{2}{p}\tau} \right)^2 = (\Phi_M)_0.$$

D'après la définition de $f(\alpha)$, on a

$$f\left(\tilde{\alpha}\right) = \Phi_M(v^+) - \Phi_M(v) \leq - \frac{-\sqrt{\frac{p}{2}} [(\Phi_M)]^{\frac{1}{2}}}{\left[\sqrt{\frac{2}{p}}p + (2+p)(p+4)\right] \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}(\Phi_M)_0\right)^{\frac{1}{p}}\right]^2}.$$

Soit K le nombre total d'itérations internes dans l'itération externe. En utilisant (3.22), on a

$$\begin{aligned} K &\leq \frac{[(\Phi_M)_0]^\gamma}{\kappa\gamma} \\ &= \frac{4\sqrt{p} + 2\sqrt{2}(2+p)(p+4)}{\sqrt{p}} \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}(\Phi_M)_0\right)^{\frac{1}{p}}\right]^2 [(\Phi_M)_0]^{\frac{1}{2}}. \end{aligned}$$

soit $(\Phi_M)_0$, défini dans (4.10), et soit $\tau \geq 1$, en utilisant (3.24) alors, le nombre total d'itérations pour avoir une solution approchée avec $n\mu < \epsilon$ est majoré par

$$\frac{4\sqrt{p} + 2\sqrt{2}(2+p)(p+4)}{\sqrt{p}} \left[1 + \log\left(1 + \frac{4}{p}\sqrt{\frac{p}{2}}(\Phi_M)_0\right)^{\frac{1}{p}}\right]^2 [(\Phi_M)_0]^{\frac{1}{2}} \frac{\log \frac{n}{\epsilon}}{\theta}.$$

Pour les méthodes à long-pas avec $\tau = \mathbf{O}(n)$ et $\theta = \Theta(1)$, nous distinguons les deux cas :

Le premier cas si $p \in [1, +\infty[$, nous obtenons pour les méthodes à long-pas $(\Phi_M)_0 = \mathbf{O}(p^2n)$ et une borne de $\mathbf{O}\left(\sqrt{np^5}(\log pn)^2 \log \frac{n}{\epsilon}\right)$ itérations.

Le deuxième cas si $p \in]0, 1[$, nous obtenons pour les méthodes à long-pas $(\Phi_M)_0 = \mathbf{O}(n)$ et une borne de $\mathbf{O}\left(\sqrt{\frac{n}{p^5}}\left(\log \frac{n}{p}\right)^2 \log \frac{n}{\epsilon}\right)$ itérations.

Pour les méthodes court-pas, on a $\tau = \mathbf{O}(1)$ et $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$, nous distinguons les deux cas :

Le premier cas si $p \in [1, +\infty[$, nous obtenons pour les méthodes à court-pas $(\Phi_M)_0 = \mathbf{O}(p^2)$ et une borne de $\mathbf{O}\left(\sqrt{np^5} \log \frac{n}{\epsilon}\right)$ itérations.

Le deuxième cas si $p \in]0, 1[$, nous obtenons pour les méthodes à court-pas $(\Phi_M)_0 = \mathbf{O}(1)$ et une borne de $\mathbf{O}\left(\sqrt{\frac{n}{p^5}} \log \frac{n}{\epsilon}\right)$ itérations.

4.2.3 Expérimentations numériques

Dans cette sous-section, nous présentons quelques résultats numériques. Nous considérons l'exemple suivant : $n = 2m$,

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

$c(i) = -1$, $c(i + m) = 0$ et $b(i) = 2$, pour $i = 1, \dots, m$.

Le point de départ (Initialisation) :

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

Nous prenons $\mu^0 = 1$, $\tau = 1$ et $\epsilon = 10^{-4}$. En outre, les paramètres m , p et θ sont pris comme suit : $m \in \{5, 15, 25, 35, 50\}$, $p \in \{0.5, 1, 4\}$ et $\theta \in \{0.1, 0.5, 0.9\}$.

Dans le tableau des résultats, (m, n) représente la taille de l'exemple, (Itr Int) représente le nombre total d'itérations internes et (Itr Ext) représente le nombre total d'itérations externes. Une étude numérique comparative est présentée dans les tableaux suivants :

tableaux comparatifs

$p = 0.5$	$\theta = 0.1$		$\theta = 0.5$		$\theta = 0.9$	
(m, n)	Itr Int	Itr Ext	Itr Int	Itr Ext	Itr Int	Itr Ext
(5, 10)	3243	117	7244	19	43395	7
(15, 30)	5297	125	14893	20	99326	7
(25, 50)	6960	129	22501	21	149611	7
(35, 70)	8474	132	29042	21	197663	7
(50, 100)	10559	135	40447	22	311676	8

$p = 1$	$\theta = 0.1$		$\theta = 0.5$		$\theta = 0.9$	
(m, n)	Itr Int	Itr Ext	Itr Int	Itr Ext	Itr Int	Itr Ext
(5, 10)	1438	115	2538	19	8641	7
(15, 30)	2408	124	4673	20	16268	7
(25, 50)	3145	128	6583	21	22241	7
(35, 70)	3802	131	8035	21	27491	7
(50, 100)	4671	134	10501	22	40309	8
$p = 4$	$\theta = 0.1$		$\theta = 0.5$		$\theta = 0.9$	
(m, n)	Itr Int	Itr Ext	Itr Int	Itr Ext	Itr Int	Itr Ext
(5, 10)	2028	112	3194	18	8893	7
(15, 30)	3545	122	5962	20	15325	7
(25, 50)	4677	127	8073	21	19900	7
(35, 70)	5619	130	9568	21	23652	7
(50, 100)	6837	133	12028	22	33136	8

Commentaires

Ces différents tests numériques pour plusieurs dimensions confirment nos résultats théoriques et consolident notre objectif d'améliorer la complexité algorithmique de l'algorithme.

4.2.4 Conclusion

Dans cette section, nous avons analysé l'algorithme de points intérieurs de type primal-dual à court et long-pas, décrit dans l'algorithme 3., qui reposent sur la fonction du noyau paramétrée (4.1). La fonction proposée est non logarithmique. Nous avons prouvé que l'itération liée à une méthode de points intérieurs à long-pas, basée sur la fonction noyau considérée dans cette section est $\mathbf{O}(\sqrt{n}(\log n)^2 \log \frac{n}{\epsilon})$ et pour les méthodes à court-pas, nous obtenons la meilleure borne connue, à savoir $\mathbf{O}(\sqrt{n} \log \frac{n}{\epsilon})$, en prenant simplement $p = \Theta(1)$.

4.3 Fonction noyau avec terme barrière trigonométrique

4.3.1 Propriétés de la fonction noyau

Dans cette partie, nous présentons une nouvelle fonction noyau paramétrée et nous donnons ses propriétés qui sont essentielles à notre analyse de la complexité.

Nous rappelons que notre fonction paramétrée indiquée dans le tableau 1 est ψ_B , définie par :

$$\psi_B(t) = \frac{t^2 - 1}{2} + \frac{4}{\pi p} [\tan^p h(t) - 1], \quad h(t) = \frac{\pi}{2t + 2}, \quad p \geq 2. \quad (5.1)$$

Nous donnons les trois premières dérivées par rapport à t :

$$\psi'_B(t) = t + \frac{4}{\pi} (1 + \tan^2 h(t)) (\tan^{p-1} h(t)) h'(t), \quad (5.2)$$

$$\psi''_B(t) = 1 + u(t) \left[\begin{array}{c} \left[\begin{array}{c} (p-1) \tan^{p-2} h(t) + \\ (p+1) \tan^p h(t) \end{array} \right] (h'(t))^2 + \\ [\tan^{p-1} h(t)] h''(t) \end{array} \right], \quad (5.3)$$

$$\psi'''_B(t) = u(t) \left[\begin{array}{c} \left[\begin{array}{c} (p-1)(p-2) \tan^{p-3} h(t) + \\ 2p^2 \tan^{p-1} h(t) + \\ (p+1)(p+2) \tan^{p+1} h(t) \end{array} \right] (h'(t))^3 + \\ \left[\begin{array}{c} 3(p-1) \tan^{p-2} h(t) + \\ 3(p+1) \tan^p h(t) \end{array} \right] h''(t) h'(t) + \\ [\tan^{p-1} h(t)] h'''(t) \end{array} \right], \quad (5.4)$$

tel que

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

et

$$\begin{aligned} h'(t) &= \frac{-\pi}{2(t+1)^2}. \\ h''(t) &= \frac{\pi}{(t+1)^3}. \\ h'''(t) &= \frac{-3\pi}{(t+1)^4}. \end{aligned} \tag{5.5}$$

Lemme 4.3.1 *Pour h , définie en (5.1), et $p \geq 2$, alors*

$$0 < h(t) < \frac{\pi}{2}, \quad t > 0. \tag{5.6}$$

$$\tan h(t) > 0, \quad t > 0. \tag{5.7}$$

$$\left[\begin{array}{c} [(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] (h'(t))^2 + \\ [\tan^{p-1} h(t)] h''(t) \end{array} \right] > 0, \quad t > 0. \tag{5.8}$$

$$\tan h(t) - \frac{2}{(p+1)\pi t} \geq 0, \quad t > 0. \tag{5.9}$$

Preuve. Pour montrer (5.6), comme h décroissante sur $]0, +\infty[$ et

$$\lim_{t \rightarrow 0^+} h(t) = \frac{\pi}{2}, \quad \lim_{t \rightarrow +\infty} h(t) = 0$$

on a

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

Pour montrer (5.7), en utilisant (5.6), nous obtenons

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

Pour montrer (5.8), en utilisant (5.5) et (5.7), pour tout $t > 0$, nous obtenons

$$\left[\begin{array}{c} [(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] (h'(t))^2 + \\ [\tan^{p-1} h(t)] h''(t) \end{array} \right] > 0.$$

Pour prouver (5.9), définissant $g(t) = (\tan h(t)) - \frac{2}{(p+1)\pi t}$. La dérivée première de g

est donnée par

$$\begin{aligned} g'(t) &= \frac{h'(t)}{(\cos h(t))^2} + \frac{2}{(p+1)\pi t^2} \\ &= \frac{1}{(p+1)\pi t^2 (\cos h(t))^2} \left((p+1) \pi t^2 h'(t) + 2 (\cos h(t))^2 \right). \end{aligned}$$

En utilisant

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

on a $\sin\left(\frac{\pi}{2} - h(t)\right) = \cos h(t)$, et $\sin\left(\frac{\pi}{2} - h(t)\right) \leq \left(\frac{\pi}{2} - h(t)\right)$, il se ensuit qu'il

$$\begin{aligned} g'(t) &= \frac{1}{(p+1)\pi t^2 (\cos h(t))^2} \left((p+1) \pi t^2 h'(t) + 2 \left(\sin\left(\frac{\pi}{2} - h(t)\right) \right)^2 \right) \\ &\leq \frac{1}{(p+1)\pi t^2 (\cos h(t))^2} \left((p+1) \pi t^2 h'(t) + 2 \left(\frac{\pi}{2} - h(t) \right)^2 \right) \\ &= \frac{-2p}{(p+1)\pi t^2 (\cos h(t))^2} \left(\frac{\pi t}{2t+2} \right)^2 < 0 \end{aligned}$$

Ainsi g est décroissante sur $]0, +\infty[$, et comme $\lim_{t \rightarrow +\infty} g(t) = 0$, ce qui implique (5.9).

Ce qui termine la preuve. ■

4.3.2 Eligibilité de la nouvelle fonction noyau

Le lemme suivant sert à prouver que la nouvelle fonction du noyau (5.1) est éligible.

Lemme 4.3.2 *Soit ψ_B , définie dans (5.1), et $t > 0$. Alors,*

$$\psi_B''(t) > 1. \tag{5.a}$$

$$\psi_B'''(t) < 0. \tag{5.b}$$

$$t\psi_B''(t) - \psi_B'(t) > 0. \tag{5.c}$$

$$t\psi_B''(t) + \psi_B'(t) > 0. \tag{5.d}$$

Preuve. Pour montrer (5.a), la dérivée seconde de ψ_B est donnée dans (5.3), en utilisant (5.5) et (5.8), pour tout $t > 0$, on a

$$\left[[(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] (h'(t))^2 + [\tan^{p-1} h(t)] h''(t) \right] > 0.$$

ce qui implique que (5.a).

Pour montrer (5.b), nous considérons (5.4), (5.5) et (5.7), on obtient

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

Pour montrer (5.c), on utilise encore (5.7) et (5.8), on a

$$t\psi_B''(t) - \psi_B'(t) = u(t) \begin{bmatrix} (h'(t))^2 [(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] t + \\ h''(t) (\tan^{p-1} h(t)) t + (\tan^{p-1} h(t)) (-h'(t)) \end{bmatrix}.$$

La forme (5.5) et la positivité de $-h'(t)$, le côté droit de la dernière égalité est positif, ce qui prouve (5.c).

Pour montrer (5.d), en utilisant (5.2) et (5.3), on a

$$t\psi_B''(t) + \psi_B'(t) = 2t + u(t) \begin{bmatrix} (h'(t))^2 [(p-1) \tan^{p-2} h(t)] t + \\ (\tan^{p-1} h(t)) \begin{bmatrix} ((p+1) (h'(t))^2 \tan h(t)) t + \\ h''(t) t + h'(t) \end{bmatrix} \end{bmatrix},$$

et

$$\begin{bmatrix} ((p+1) (h'(t))^2 \tan h(t)) t + \\ h''(t) t + \\ h'(t) \end{bmatrix} = \frac{4\pi}{(2t+2)^4} \begin{bmatrix} 2t^2 + \\ \pi (p+1) t \left(\tan h(t) - \frac{2}{(p+1)\pi t} \right) \end{bmatrix}$$

en utilisant (5.9), le côté droit de l'égalité ci-dessus est positif, ce qui prouve (5.d).

Ce qui termine la preuve. ■

La dernière propriété (5.d), est équivalente que la fonction $t \mapsto \psi_B(e^t)$ est convexe et

$$\psi_B(\sqrt{t_1 t_2}) \leq \frac{1}{2}(\psi_B(t_1) + \psi_B(t_2))$$

pour tous $t_1, t_2 \geq 0$. Cette propriété est connue dans le lemme **3.4.2**.

Nous présentons dans la suite quelques propriétés et résultats techniques concernant la nouvelle fonction noyau.

Lemme 4.3.3 *Pour ψ_B , nous avons :*

$$\frac{1}{2}(t-1)^2 \leq \psi_B(t) \leq \frac{1}{2}[\psi'_B(t)]^2, \quad t > 0. \quad (5.10)$$

$$\psi_B(t) \leq \left(1 + \frac{\pi}{8}p\right)(t-1)^2, \quad t > 1. \quad (5.11)$$

Preuve. Pour montrer (5.10), en utilisant (5.a), on a

$$\psi_B(t) = \int_1^t \int_1^x \psi''_B(y) dy dx \geq \int_1^t \int_1^x dy dx = \frac{1}{2}(t-1)^2$$

$$\begin{aligned} \psi_B(t) &= \int_1^t \int_1^x \psi''_B(y) dy dx \\ &\leq \int_1^t \int_1^x \psi''_B(y) \psi''_B(x) dy dx \\ &= \int_1^t \psi''_B(x) \psi'_B(x) dx \\ &= \int_1^t \psi'_B(x) d\psi'_B(x) \\ &= \frac{1}{2}[\psi'_B(t)]^2. \end{aligned}$$

Pour montrer (5.11), comme $\psi_B(1) = \psi'_B(1) = 0$, $\psi'''_B(t) < 0$, $\psi''_B(1) = (2 + \frac{\pi}{4}p)$, et en utilisant le développement de Taylor, on obtient

$$\begin{aligned} \psi_B(t) &= \psi_B(1) + \psi'_B(1)(t-1) + \frac{1}{2}\psi''_B(1)(t-1)^2 + \frac{1}{6}\psi'''_B(\xi)(\xi-1)^3 \\ &= \frac{1}{2}\psi''_B(1)(t-1)^2 + \frac{1}{6}\psi'''_B(\xi)(\xi-1)^3 \\ &\leq \frac{1}{2}\psi''_B(1)(t-1)^2 \\ &= \left(1 + \frac{\pi}{8}p\right)(t-1)^2. \end{aligned}$$

pour certains ξ vérifiant : $1 \leq \xi \leq t$.

Ce qui termine la preuve. ■

Soit $\varrho : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de ψ_B pour $t \geq 1$ et soit $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de $\frac{-1}{2}\psi'_B$ pour tout $t \in]0, 1]$. Alors nous avons le lemme suivant :

Lemme 4.3.4 *Pour ψ_B , on a :*

$$1 + \sqrt{\frac{8}{8 + \pi p}}s \leq \sigma(s) \leq 1 + \sqrt{2s}, \quad s \geq 0. \quad (5.12)$$

$$\tan h(t) \leq (4z + 2)^{\frac{1}{p+1}}, \quad z \geq 0. \quad (5.13)$$

Preuve. Pour montrer (5.12), soit

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

En utilisant (5.10), on a

$$\psi_B(t) \geq \frac{1}{2}(t-1)^2.$$

alors

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

ce qui implique que

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

En utilisant (5.11), on a

$$s = \psi_B(t) \leq \left(1 + \frac{\pi}{8}p\right)(t-1)^2, \quad t > 1,$$

donc

$$t = \sigma(s) \geq 1 + \sqrt{\frac{8}{8 + \pi p}}s.$$

Pour montrer (5.13), soit

$$z = \frac{-1}{2} \psi'_B(t), t \in]0, 1] \Leftrightarrow 2z = -\psi'_B(t), t \in]0, 1].$$

En utilisant la définition de

$$\rho : \rho(z) = t, t \in]0, 1].$$

et en utilisant la définition de $\psi'_B(t)$ et $h'(t)$, on a

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

ce qui implique

$$\begin{aligned} (1 + \tan^2 h(t)) (\tan^{p-1} h(t)) &= (2z + t) \frac{\pi}{4(-h'(t))} \\ &\leq 2(2z + 1) \text{ for } t \in]0, 1] \end{aligned}$$

et

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

ce qui implique que

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

Ce qui termine la preuve. ■

Lemme 4.3.5 Soit $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$, Si $\Phi_B(v) \leq \tau$, alors on a

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

Preuve. Comme $\frac{1}{\sqrt{1-\theta}} \geq 1$ et $\varrho\left(\frac{\Phi_B(v)}{n}\right) \geq 1$, nous obtenons $\frac{\varrho\left(\frac{\Phi_B(v)}{n}\right)}{\sqrt{1-\theta}} \geq 1$.

Et pour $t \geq 1$, on a

$$\psi_B(t) \leq \frac{t^2 - 1}{2}.$$

En utilisant le Théorème **3.4.1**, avec $\beta = \frac{1}{\sqrt{1-\theta}}$, l'inégalité (5.12) et $\Phi_B(v) \leq \tau$, on obtient

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

Ce qui termine la preuve. ■

Soit

$$(\Phi_B)_0 = \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)} = L(n, \theta, \tau), \quad (5.14)$$

alors, $\Phi_B(v_+)$ majoré par $(\Phi_B)_0$ dans l'algorithme.

Lemme 4.3.6 *Soit $\delta(v)$, défini dans (3.12). Alors, on a*

$$\delta(v) \geq \sqrt{\frac{1}{2}\Phi_B(v)}.$$

Preuve. En utilisant (5.10), on a

$$\Phi_B(v) = \sum_{i=1}^n \psi_B(v_i) \leq \sum_{i=1}^n \frac{1}{2} (\psi'_B(v_i))^2 = \frac{1}{2} \|\nabla \Phi_B(v)\|^2 = 2\delta(v)^2,$$

donc

$$\delta(v) \geq \sqrt{\frac{1}{2}\Phi_B(v)}.$$

Ce qui termine la preuve. ■

4.3.3 Analyse de la complexité

Tout au long de cette partie, nous supposons que $\tau \geq 1$. En utilisant le lemme 4.3.6 et l'hypothèse que $\Phi_B(v) \geq \tau$, on a

$$\delta(v) \geq \sqrt{\frac{1}{2}}.$$

D'après les lemmes 3.4.6-3.4.8 , on a

Lemme 4.3.7 *Soit $\bar{\alpha}$, la valeur définie dans le lemme 3.4.8. Si*

$$\Phi_B = \Phi_B(v) \geq \tau \geq 1,$$

alors on a

$$\bar{\alpha} \geq \frac{1}{(9 + 4p\pi)(8\delta + 2)^{\frac{p+2}{p+1}}}$$

Preuve. En utilisant le lemme 3.4.9, la définition de $\psi_B''(t)$, et (5.13) on a

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

et pour $\rho(2\delta) = t \in]0, 1]$, on a

$$\begin{aligned} (h'(t))^2 &= \frac{\pi^2}{4(t+1)^4} \leq \left(\frac{\pi}{2}\right)^2 \\ h''(t) &= \frac{\pi}{(t+1)^3} \leq \pi \end{aligned}$$

$$\psi_B''(t) = 1 + u(t) \left[\begin{array}{c} [(p-1)\tan^{p-2}h(t) + (p+1)\tan^p h(t)](h'(t))^2 + \\ [\tan^{p-1}h(t)]h''(t) \end{array} \right],$$

pour $z = 2\delta$ on a $\tan h(t) \leq (4z + 2)^{\frac{1}{p+1}}$, ce qui implique

$$\begin{aligned}\tan^2 h(\rho(2\delta)) &\leq (8\delta + 2)^{\frac{2}{p+1}} \\ \tan^{p-2} h(\rho(2\delta)) &\leq (8\delta + 2)^{\frac{p-2}{p+1}} \\ \tan^{p-1} h(\rho(2\delta)) &\leq (8\delta + 2)^{\frac{p-1}{p+1}} \\ \tan^p h(\rho(2\delta)) &\leq (8\delta + 2)^{\frac{p}{p+1}}\end{aligned}$$

et $(8\delta + 2) > 1$ ce qui implique que

$$\psi_B''(t) \leq \left[1 + \frac{4}{\pi} (2) \left(2p \left(\frac{\pi}{2} \right)^2 + \pi \right) \right] (8\delta + 2)^{\frac{p+2}{p+1}}.$$

on a

$$\begin{aligned}\bar{\alpha} &\geq \frac{1}{\left[1 + \frac{4}{\pi} (2) \left(2p \left(\frac{\pi}{2} \right)^2 + \pi \right) \right] (8\delta + 2)^{\frac{p+2}{p+1}}}, \\ &= \frac{1}{(9 + 4p\pi)(8\delta + 2)^{\frac{p+2}{p+1}}}.\end{aligned}$$

Ce qui termine la preuve. ■

Soit

$$\tilde{\alpha} = \frac{1}{(9 + 4p\pi)(8\delta + 2)^{\frac{p+2}{p+1}}}. \quad (5.15)$$

$\tilde{\alpha}$ est le pas de déplacement et $\tilde{\alpha} \leq \bar{\alpha}$.

Lemme 4.3.8 *On considère $\tilde{\alpha}$ le pas de déplacement, défini dans (4.15), et soit*

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

Alors

$$f(\tilde{\alpha}) \leq -\frac{\sqrt{2}}{288(13p + 9)} [\Phi_B(v)]^{\frac{p}{2(p+1)}}. \quad (5.16)$$

Preuve. En utilisant le lemme **3.4.11** pour $\alpha = \tilde{\alpha}$ et (5.14), on a

$$\begin{aligned}
f(\tilde{\alpha}) &\leq -\tilde{\alpha}\delta^2 \\
&= -\frac{\delta^2}{(9+4p\pi)(8\delta+2)^{\frac{p+2}{p+1}}} \\
&\leq -\frac{\delta^2}{(9+4p\pi)(8\delta+2(2\delta))^{\frac{p+2}{p+1}}} \\
&\leq -\frac{\delta^{\frac{p}{p+1}}}{144(13p+9)} \\
&\leq -\frac{\sqrt{2}}{288(13p+9)} [\Phi_B(v)]^{\frac{p}{2(p+1)}}
\end{aligned}$$

car

$$\delta \geq \sqrt{\frac{1}{2}\Phi_B(v)} \geq \sqrt{\frac{1}{2}}.$$

Ce qui termine la preuve. ■

Après la mise à jour de μ à $(1-\theta)\mu$, on a

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

D'après la définition de $f(\alpha)$, on a

$$f(\tilde{\alpha}) = \Phi_B(v^+) - \Phi_B(v) \leq -\frac{\sqrt{2}}{288(13p+9)} [(\Phi_B)]^{\frac{p}{2(p+1)}}.$$

Soit K le nombre total d'itérations internes dans l'itération externe. En utilisant (3.22), on a

$$\begin{aligned}
K &\leq \frac{[(\Phi_B)_0]^\gamma}{\kappa\gamma} \\
&= \left(\frac{288(13p+9)(p+1)\sqrt{2}}{(p+2)} \right) (\Phi_B)_0^{\frac{p+2}{2(p+1)}}.
\end{aligned}$$

Soient $(\Phi_B)_0$ la borne supérieure définie dans (5.14) et $\tau \geq 1$. En utilisant (3.24) alors, le nombre total d'itérations pour avoir une solution approchée avec $n\mu < \epsilon$ est

majoré par

$$\left(\frac{288 (13p + 9) (p + 1) \sqrt{2}}{(p + 2)} \right) (\Phi_B)_0^{\frac{p+2}{2(p+1)}} \frac{\log \frac{n}{\epsilon}}{\theta}. \quad (5.17)$$

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

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

Dans le cas d'une méthode à court-pas, on a $\tau = \mathbf{O}(1)$ et $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$. Le remplacement de ces valeurs dans (5.17) nous donne la meilleure limite possible. Une meilleure borne est obtenue comme suit :

En utilisant (5.11) et comme

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

on a

$$\begin{aligned} \Phi_B(v_+) &\leq n\psi_B\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_B(v)}{n}\right)\right) \\ &\leq n\left(1 + \frac{\pi}{8}p\right)\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi_B(v)}{n}\right) - 1\right)^2 \\ &= \frac{n(1+\frac{\pi}{8}p)}{(1-\theta)}\left(\sigma\left(\frac{\Phi_B(v)}{n}\right) - \sqrt{1-\theta}\right)^2 \\ &\leq \frac{n(1+\frac{\pi}{8}p)}{(1-\theta)}\left(\left(1 + \sqrt{2\frac{\Phi_B(v)}{n}}\right) - \sqrt{1-\theta}\right)^2 \\ &= \frac{n(1+\frac{\pi}{8}p)}{(1-\theta)}\left((1 - \sqrt{1-\theta}) + \sqrt{2\frac{\Phi_B(v)}{n}}\right)^2 \\ &\leq \frac{n(1+\frac{\pi}{8}p)}{(1-\theta)}(\theta + \sqrt{2\frac{\tau}{n}})^2 \\ &= \frac{(1+\frac{\pi}{8}p)}{(1-\theta)}(\theta\sqrt{n} + \sqrt{2\tau})^2 = (\Phi_B)_0. \end{aligned}$$

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

En utilisant cette borne supérieure $(\Phi_B)_0$, nous obtenons un majorant de nombre total d'itérations suivant :

$$\left(\frac{288 (13p + 9) (p + 1) \sqrt{2}}{(p + 2)} \right) (\Phi_B)_0^{\frac{p+2}{2(p+1)}} \frac{\log \frac{n}{\epsilon}}{\theta}.$$

Alors $(\Phi_B)_0 = \mathbf{O}(p)$, et la complexité algorithmique devient

$$\mathbf{O}\left(p^2\sqrt{n}\log\frac{n}{\epsilon}\right) \text{ iterations.}$$

Nous remarquons que tous les résultats théoriques de cette fonction obtenus pour $p \geq 2$ restent valables pour $p = 1$.

4.3.4 Expérimentations numériques et étude comparative

Pour prouver l'efficacité de notre nouvelle fonction noyau et évaluer son influence sur le comportement de l'algorithme, nous avons effectué des tests numériques comparatifs entre les trois fonctions noyau, avec un terme barrière trigonométrique, suivantes :

1) Première fonction noyau, donnée par M. Peyghami et al. dans [42], et noté fonction noyau 8.

2) Deuxième fonction noyau, donnée par M. Peyghami et al. dans [41], et noté fonction noyau 9.

3) Notre nouvelle fonction noyau, notée fonction noyau B .

Les exemples suivants sont tirés de la littérature (voir par exemple la référence [28]). Pour résoudre le problème numériquement, nous avons utilisé le logiciel Dev Pascal. Nous avons pris $\epsilon = 10^{-4}$, $\theta = 0.99$, $\tau = 10$ pour des exemples de taille fixe et $\tau = n$ pour des exemples avec une taille variable.

Nous choisissons α le pas du déplacement, nous prenons $0 < \alpha < \bar{\alpha}$

- 1) Pour la fonction noyau 8, $\alpha = \frac{\delta^{\frac{4}{3}}}{5020}$.
- 2) Pour la fonction noyau 9, $\alpha = \frac{1}{90\delta\left(1+\frac{1}{3}\log(4\delta+1)\right)^2}$.
- 3) Pour la fonction noyau B , $\alpha = \frac{1}{(9+8\pi)(8\delta+2)^{\frac{4}{3}}}$, $p = 2$.

Dans les tableaux des résultats, (m, n) représente la taille de l'exemple, (Itr) représente le nombre d'itérations externes nécessaires pour déterminer la solution optimale, (temps (s)) représente le temps de calcul. Nous résumons ces études numériques dans les tableaux comparatifs suivants, pour respectivement des exemples avec la taille fixe et variable.

– **Exemples avec une taille fixe**

Nous considérons les exemples suivants :

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & -3 \end{pmatrix},$$

$$b = (1, 0.5)^t, c = (1, 2, 3, 4)^t.$$

Le point de départ (Initialisation) :

$$x^0 = (0.5, 0.27, 0.14, 0.09)^t,$$

$$s^0 = (1, 2, 3, 4)^t, y^0 = (0, 0)^t.$$

$$A = \begin{pmatrix} 2 & -1 & 1 & 1 \\ 1 & 1 & -1 & 2 \\ 1 & 1 & 1 & 2 \end{pmatrix},$$

$$b = (3, 4, 5)^t, c = (3, 2, 1, 3)^t.$$

Le point de départ (Initialisation) :

$$x^0 = (1.3941, 0.5368, 1.1970, 0.9361)^t,$$

$$s^0 = (2, 2, 3, 3)^t, y^0 = (0, 1, -1)^t.$$

$$A = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

$$b = (8, 7, 3)^t, c = (-4, -5, 0, 0, 0)^t.$$

Le point de départ (Initialisation) :

$$x^0 = (2.2534, 1.5743, 1.9185, 1.5976, 1.4256)^t,$$

$$s^0 = (1, 3, 2, 2, 2)^t, y^0 = (-2, -2, -2)^t.$$

$$A = \begin{pmatrix} -1 & 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 2 & -3 & 2 & 0 & 1 & 0 \\ -3 & 2 & 1 & 0 & 0 & 0 & 1 \\ 3 & 5 & 4 & 0.5 & 0 & 0 & 0 \end{pmatrix},$$

$$b = (1, 2, 0, 2)^t, c = (1, 1, 0, 0, 1, 1, -2)^t.$$

Le point de départ (Initialisation) :

$$x^0 = (0.2451, 0.1174, 0.1057, 0.5101, 1.5101, 1.5321, 1.0621, 0.3947)^t,$$

$$s^0 = (1, 22, 15, 1.5, 1, 1, 1)^t, y^0 = (0, 0, -3, -3)^t.$$

$$A = \begin{pmatrix} 1 & 0 & -4 & 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 3 & 1 & 0 & -1 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & 5 & -3 & 3 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 & 1 & -5 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2 & 1 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & -3 & 2 & -1 & 4 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$b = (1, 4, 4, 5, 7, 5)^t, c = (-4, -5, -1, -3, 5, -8, 0, 0, 0, 0, 0, 0)^t.$$

Le point de départ (Initialisation) :

$$x^0 = (0.1547, 0.4469, 1.6113, 1.9393, 0.6864, 0.1696, 0.6170, 0.4531, 2.7393, 1.7290, 1.5998, 1.1550)^t,$$

$$s^0 = (2, 22, 2, 33, 38, 37, 3, 3, 3, 3, 15, 3)^t, y^0 = (-3, -3, -3, -3, -15, -3)^t.$$

tableau comparatif

(m, n)	fonction noyau 8		fonction noyau 9		fonction noyau B	
	Itr	temps(s)	Itr	temps(s)	Itr	temps(s)
(2, 4)	4	3.54	4	0.44	4	0.16
(3, 4)	4	7.49	4	0.76	4	0.31
(3, 5)	4	11.76	4	1.08	4	0.48
(4, 7)	4	23.15	4	1.73	4	0.88
(6, 12)	4	127.71	4	6.86	4	4.37

Comparaison des exemples avec la taille fixe.

– **Exemple de taille variable**

Nous considérons l'exemple suivant : $n = 2m$,

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

$c(i) = -1$, $c(i + m) = 0$ et $b(i) = 2$, pour $i = 1, \dots, m$.

Le point de départ (Initialisation) :

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

tableau comparatif

(m, n)	fonction noyau 8		fonction noyau 9		fonction noyau B	
	Itr	temps(s)	Itr	temps(s)	Itr	temps(s)
(5, 10)	4	140.85	4	6.80	4	2.48
(10, 20)	4	833.61	4	35.30	4	25.49
(15, 30)	5	4331.32	5	169.55	5	142.41
(25, 50)	—	—	5	986.18	5	867.49
(50, 100)	—	—	5	11394.10	5	10550.14

Comparaison des exemples avec une taille variable.

– **Commentaires :**

Les résultats numériques obtenus sur toutes les instances utilisées montrent l'efficacité de notre nouvelle fonction noyau. Nous notons que, lorsque la dimension du problème devient grande, la différence entre la nouvelle fonction noyau et celles proposées par M. R. Peyghami et al. (fonction noyau 8 et fonction noyau 9) devient très grande en termes de temps de calcul. Ces résultats numériques consolident et confirment nos résultats théoriques.

Remarque 4.3.1 *Les résultats numériques obtenus dans les deux chapitres 2, 4 ont montré que la méthode projective est plus rapide en termes de temps d'exécution par rapport à la méthode de trajectoire centrale via une fonction noyau. Ce résultat est tout à fait normal car les deux méthodes ne sont pas de même nature : la première méthode ne peut fournir qu'une solution primale tandis que la deuxième méthode donne une solution primale-duale.*

4.3.5 Conclusion

Notre objectif dans ce travail est de proposer une nouvelle fonction noyau pour améliorer la complexité algorithmique de l'algorithme de points intérieurs.

Dans cette partie, nous avons analysé les méthodes à court et long-pas de l'algorithme de points intérieurs primal-dual décrit dans l'algorithme **3.**, qui est basé sur la nouvelle fonction noyau avec un terme barrière trigonométrique (5.1). Nous avons montré que la borne de l'itération liée à une méthode à long-pas basée sur la fonction noyau considérée est $\mathbf{O}\left(pn^{\frac{p+2}{2(p+1)}} \log \frac{n}{\epsilon}\right)$. Fixant $p \geq 2$, cette borne améliore tous les résultats, concernant la complexité des méthodes à long-pas, obtenus jusqu'à présent sur la base de la fonction noyau trigonométrique donnée par M. R. Peyghami et al. (voir [42]), i.e., $\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$. Un autre choix est intéressant si p dépend de n , c'est la valeur minimisant la complexité de l'itération. En particulier, en prenant $p = \left(\frac{\log n}{2} - 1\right)$, nous obtenons la meilleure complexité pour les méthodes à long-pas à savoir $\mathbf{O}\left(\sqrt{n} (\log n) \log \frac{n}{\epsilon}\right)$. Cette borne améliore tous les résultats, concernant la complexité des méthodes à long-pas, obtenus jusqu'à présent sur la base de la fonction noyau trigonométrique donnée par M. R. Peyghami et al. (voir [41]), i.e., $\mathbf{O}\left(n (\log n)^2 \log \frac{n}{\epsilon}\right)$. Pour les méthodes à court-pas, nous obtenons la meilleure borne de complexité, à savoir $\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$, en prenant simplement $p = \mathbf{O}(1)$. Ces résultats représentent une contribution importante pour améliorer la complexité algorithmique du problème étudié.

Conclusion générale

Dans cette thèse, nous avons étudié le problème de complexité algorithmique dans les deux familles des méthodes de points intérieurs les plus connues en optimisation :

- La réduction du potentiel projective (algorithme de Karmarkar).
- La trajectoire centrale (**TC**).

Nous avons proposé deux nouveaux pas de déplacement pour accélérer la convergence de l'algorithme de Karmarkar et améliorer sa complexité algorithmique. Le premier pas proposé est meilleur que celui introduit par Schrijver [47, 7], et le deuxième représente le meilleur pas de déplacement fixe obtenu jusqu'à présent [12]. Ces nouvelles améliorations sont d'ordre théorique, algorithmique et numérique.

D'autre part, nous nous sommes intéressés à la forme analytique de la fonction noyau, dans l'algorithme de trajectoire centrale via une fonction noyau pour résoudre un problème linéaire. Nous avons proposé une fonction noyau avec paramètre qui généralise la complexité algorithmique obtenue par Y. Q. Bai et al. dans [3]. De même, nous avons proposé une nouvelle fonction noyau avec terme barrière trigonométrique qui est la première fonction de ce type donnant la meilleure complexité algorithmique, obtenue jusqu'à présent, pour les méthodes à long-pas (voir [11]). Cette fonction, avec les différentes valeurs de son paramètre, généralise les complexités algorithmiques trouvées par différents chercheurs (voir [19, 3, 42, 41, 30, 33]). Ces propositions représentent des nouvelles contributions d'ordre algorithmique, théorique et numérique.

Les résultats obtenus sont encourageants, ce qui motivent à généraliser nos travaux à plusieurs problèmes d'optimisation tels que le problème semi-défini (*SDP*), les problèmes de complémentarité linéaire, les problèmes de complémentarité semi-défini et les problèmes convexes avec contraintes linéaires.

Annexes

Annexe 1 : Calcul de la projection dans l'algorithme de Ye-Lustig

L'étape 1 dans l'algorithme de Ye-Lustig est la plus dominante. En effet, le calcul de la projection est l'opération la plus coûteuse en volume calculatoire, car il nécessite le calcul de l'inverse de la matrice $(B_k B_k^t)$ à chaque itération. A cet effet, l'efficacité de l'algorithme dépend en grande partie de la manière économique du calcul de cette projection.

Nous avons :

$$P_k = \left[I - B_k^t (B_k B_k^t)^{-1} B_k \right] \begin{pmatrix} D_k c \\ -c^t x^k \end{pmatrix}$$

Posons :

$$u_k = (B_k B_k^t)^{-1} B_k \begin{pmatrix} D_k c \\ -c^t x^k \end{pmatrix}$$

qui est équivalent à résoudre le système linéaire symétrique défini positif suivant :

$$(B_k B_k^t) u_k = B_k \begin{pmatrix} D_k c \\ -c^t x^k \end{pmatrix}$$

la méthode la plus convenable pour ce système est la méthode directe de Cholesky pour les tailles moyennes et la méthode de gradient conjuguée pour les systèmes linéaires de grandes tailles.

Annexe 2 : Calcul d'une solution initiale primale strictement réalisable

Un point strictement réalisable de (PL) est une solution du problème :

$$(P_1) \begin{cases} Ax = b \\ x > 0 \end{cases}$$

Or, le problème (P_1) est équivalent au problème de minimisation :

$$(P_2) \begin{cases} \min \lambda \\ Ax + \lambda(b - A\theta) = b \\ x > 0, \lambda \geq 0, \end{cases}$$

où $\theta \in \mathbb{R}^n / \theta > 0$ est choisi arbitrairement et λ une variable artificielle.

Le problème (P_2) est un programme linéaire sous forme standard suivant :

$$(P_2) \begin{cases} \min c_1^t x_1 \\ A_1 x_1 = b_1 \\ x_1 \geq 0, \end{cases}$$

où

- $c_1 = (0, 0, 0, \dots, 1)^t \in \mathbb{R}^{n+1}$
- $x_1 = (x, \lambda) \in \mathbb{R}^{n+1}$
- $A_1 = [A, b - A\theta] \in \mathbb{R}^{m \times (n+1)}$
- $b_1 = b \in \mathbb{R}^m$

Le problème (P_2) admet une solution strictement réalisable triviale $(\theta, 1) \in \mathbb{R}^{n+1}$ tel que $\theta > 0$ arbitraire.

La résolution du problème (P_2) se ramène à celle de (P_1) et vice versa. Autrement dit si (x^*, λ^*) est une solution optimale de (P_2) alors x^* est une solution de (P_1) . Plus précisément Karmarkar démontre le théorème suivant :

Théorème 4.3.1 [28] $\exists \epsilon_0 > 0$ tel que : les deux propositions suivantes sont équivalentes :

- (P_1) est réalisable.
- (P_2) admet une solution optimale (x^*, λ^*) telle que $\lambda^* \leq \epsilon_0$.

Donc pour trouver une solution strictement réalisable pour (PL) , il suffit de résoudre le problème (P_2) par la phase 2 de l'algorithme de Ye-Lustig .

Annexe 3 : Calcul d'une solution initiale duale strictement réalisable

Un point strictement réalisable de (DL) est une solution du problème :

$$(D_1) \begin{cases} A^t y + s = c \\ y \in \mathbb{R}^m, s > 0 \end{cases}$$

On pose : $y = y^+ - y^-$ tel que $y^+ \geq 0$ et $y^- \geq 0$.

Le problème (D_1) est équivalent au problème suivant :

$$(D_2) \begin{cases} A^t y^+ + A^t y^- + s = c \\ y^+ > 0, y^- > 0, s > 0 \end{cases}$$

Or, le problème (D_2) est équivalent au problème de minimisation :

$$(D_3) \begin{cases} \min \beta \\ Mw + \beta (c - M\xi) = c \\ w > 0, \beta \geq 0 \end{cases}$$

où $\xi \in \mathbb{R}_{++}^{2m+n}$ est choisi arbitrairement et ξ une variable artificielle.

Le problème (D_3) est un programme linéaire sous la forme standard suivante :

$$(D_3) \left\{ \begin{array}{l} \min \bar{c}^t w_1 \\ Dw_1 = c \\ w_1 > 0, \beta \geq 0 \end{array} \right.$$

où

- $\bar{c} = (0, 0, 0, \dots, 1)^t \in \mathbb{R}^{2m+n+1}$
- $w_1 = (w, \beta) \in \mathbb{R}^{2m+n+1}$
- $w = (y^+, y^-, s) \in \mathbb{R}^{2m+n}$
- $D = [M, c - M\xi] \in \mathbb{R}^{n \times (2m+n+1)}$
- $M = [A^t, A^t, I] \in \mathbb{R}^{n \times (2m+n)}$

Le problème (D_3) admet une solution strictement réalisable triviale $(\xi, 1) \in \mathbb{R}^{2m+n+1}$ tel que $\xi > 0$ arbitraire.

Donc pour trouver une solution strictement réalisable pour (DL) , il suffit de résoudre le problème (D_3) par la phase 2 de l'algorithme de Ye-Lustig .

Annexe 4 : Calcul des pas de déplacement variables dans l'algorithme de Ye-Lustig

Sur le plan numérique, on suggère plutôt des pas variables α_k à chaque itération k , moyennant les méthodes de recherche linéaire sur la fonction potentiel.

$$\varphi(\alpha_k) = \min_{0 < \alpha < \alpha_{\max}} f\left(\frac{e_n}{n} - \alpha r d_k\right)$$

Ces méthodes s'avèrent très coûteuses en volume calculatoire. Dans le même ordre d'idée A. Keraghel[28] a proposé une variante plus simple qui offre des pas α_k variables (généralement $\alpha_k > 1$) d'une manière beaucoup plus facile que les recherches linéaires. Il

s'agit à chaque itération de chercher α_k qui vérifie les deux conditions suivantes :

$$\begin{aligned} \frac{e_n}{n} - \alpha_k r d_k &> 0 \quad (\text{faisabilité stricte}) \\ c^t x^{k+1} &< c^t x^k \quad (\text{monotonie}) \end{aligned}$$

Signalons que l'étude théorique de la convergence polynomiale moyennant des pas variables n'est pas établie, malgré le succès numérique des algorithmes appropriés.

Annexe 5 : Calcul des pas de déplacement variables dans la méthode (TC)

On introduit une procédure moins coûteuse que la recherche linéaire introduite par A. Keraghel [28] moyennant les conditions de la stricte positivité suivantes :

$$\begin{cases} x + \alpha_x^+ dx > 0 \\ s + \alpha_s^+ ds > 0. \end{cases}$$

Ce qui donne :

$$\alpha_x^+ = \beta \alpha_x \text{ et } \alpha_s^+ = \beta \alpha_s,$$

tel que $0 < \beta < 1$, où :

$$\alpha_x = \begin{cases} \min \left(-\frac{x_i}{dx_i} \right) & \text{avec } i \in I = \{i : dx_i < 0\} \\ 1 & \text{ailleurs} \end{cases}$$

$$\alpha_s = \begin{cases} \min \left(-\frac{s_i}{ds_i} \right) & \text{avec } i \in I = \{i : ds_i < 0\} \\ 1 & \text{ailleurs} \end{cases}$$

On prend $\alpha_k = \min(\alpha_x, \alpha_s)$. Alors le nouvel itéré est :

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

Annexe 6 : Estimation de la décroissance de la fonction barrière dans la méthode de Trajectoire Centrale via une fonction noyau

Pour expliquer la décroissance de la fonction barrière dans la méthode de trajectoire centrale via une fonction noyau, nous présentons le lemme suivant :

Lemme 4.3.9 *Soit ψ une fonction noyau satisfaisant les conditions (3.a) et (3.d), alors*

l'estimation de $\Phi_{k+1} - \Phi_k$ est décroissante par rapport à δ .

Preuve. Supposons que ψ une fonction noyau satisfaisant les conditions (3.a) et (3.d), donc on a

$$\begin{aligned}\Phi_{k+1} - \Phi_k &\leq f(\tilde{\alpha}) \leq -\frac{\delta^2}{\psi''(\rho(2\delta))}, \\ 2[\psi''(t)]^2 - \psi'(t)\psi'''(t) &> 0, \forall t < 1.\end{aligned}$$

Posons $t = \rho(2\delta)$ donc

$$t \leq 1, \text{ et } \delta = \frac{-1}{4}\psi'(t),$$

alors

$$-\frac{\delta^2}{\psi''(\rho(2\delta))} = g(t) = -\frac{[\psi'(t)]^2}{16\psi''(t)}, \forall t < 1,$$

et on a

$$g'(t) = -\frac{2[\psi''(t)]^2 - \psi'(t)\psi'''(t)}{16[\psi''(t)]^2} > 0, \forall t < 1,$$

Donc g est une fonction décroissante par rapport à t , pour $t < 1$. Comme la fonction définie par ρ est croissante par rapport à $\delta > 0$, alors la fonction ϕ définie par $\phi(\delta) = g(\rho(2\delta))$ est décroissante par rapport à $\delta > 0$.

Ce qui termine la preuve. ■

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] Y. Q. Bai, M. El Ghami, C. Roos, *A new efficient large-update primal-dual interior point method based on a finite barrier*, SIAM Journal on Optimization, 13, 766–782, (2003).
- [3] Y. Q. Bai, M. El Ghami, C. Roos, *A comparative study of kernel functions for primal-dual interior point algorithms in linear optimization*, SIAM Journal on Optimization, 15, 101–128, (2004).
- [4] Y. Q. Bai, G. Lesaja, C. Roos, G. Q. Wang, M. El Ghami, *A class of large-update and small-update primal-dual interior point algorithms for linear optimization*, Journal of Optimization Theory and Applications, 138, 341–359, (2008).
- [5] Y. Q. Bai, C. Roos, *A primal-dual interior point method based on a new kernel function with linear growth rate*, in : Proceedings of the 9th Australian Optimization Day, Perth, Australia, (2002).
- [6] Y. Q. Bai, C. Roos, *A polynomial-time algorithm for linear optimization based on a new simple kernel function*, Optimization Methods and Software, 18, 631–646, (2003).

- [7] D. Benterki, M. Bouafia, *Improving complexity of Karmarkar's approach for linear programming*, Journal of Numerical Analysis and Approximation Theory, 43(2), 159–167, (2015).
- [8] M. Bierlaire, *Introduction à l'optimisation différentiable*, press polytechniques et universitaires romandes, (2006).
- [9] M. Bouafia, D. Benterki, A. Yassine, *A Numerical Implementation of an Interior Point Methods for Linear Programming Based on a New Kernel Function*, Book chapter : Modelling, Computation and Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing, 359, 357–368, Springer-Verlag, (2015).
- [10] M. Bouafia, D. Benterki, A. Yassine, *Complexity analysis of interior point methods for linear programming based on a parameterized kernel function*, à paraître dans RAIRO-Operations Research, (2016).
- [11] M. Bouafia, D. Benterki, A. Yassine, *An efficient primal-dual Interior Point Method for linear programming problems based on a new kernel function with a trigonometric barrier term*, à paraître dans Journal of Optimization Theory and Applications, (2016).
- [12] M. Bouafia, D. Benterki, A. Yassine, *A new efficient short-step projective interior point method for linear programming*, Soumise à Ukrainian Mathematical Journal, (2014).
- [13] X. Z. Cai, G. Q. Wang, M. El Ghami, and Y. J. Yue, *Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term*, Abstract and Applied Analysis, 11 pages, Art. ID 710158, (2014).
- [14] G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, (1985).

- [15] G. B. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, In Tj.C. Koopmans, éditeur, Activity Analysis of Production and Allocation, 339–347, Wiley, NewYork,.554–567, (1951).
- [16] Zs. Darvay, *A weighted-path-following method for linear optimization*, Studia Universitatis Babes-Bolyai. Series Informatica, 47(1), 3-12, (2002).
- [17] K. R. Frisch, *The logarithmic potential method of convex programming*, Technical report, University Institute of Economics, Oslo, Norway, (1955).
- [18] R. Fletcher, *Pratical methods of optimisation*, A Wiley interscience publication, JOHN WILEY & SONS Ltd, (1991).
- [19] M. El Ghami, Z. A. Guennoun, S. Bouali, T. Steihaug, *Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term*, Jornal of Computational and applied Mathematics, 236, 3613–3623, (2012).
- [20] M. El Ghami, I. D. Ivanov, C. Roos, T. Steihaug, *A polynomial-time algorithm for LO based on generalized logarithmic barrier functions*, International Journal of Applied Mathematics, 21, 99–115, (2008).
- [21] M. El Ghami, C. Roos, *Generic primal-dual interior point methods based on a new kernel function*, RAIRO-Operations Research, 42, 199–213, (2008).
- [22] C. C. Gonzaga, *Path following methods for linear programming*, SIAM Review, 34, 167–227, (1992).
- [23] D. Hertog, C. Roos, *A survey of search directions in interior point methodes for linear programming*, Mathematical Programming, 52, 481–509, (1991).
- [24] P. Huard, *Resolution of mathematical programming with nonlinear constraints by the method of centers*, In J. Abadie, editor, Nonlinear Programming, pages 207–219. North Holland, Amsterdam, The Netherlands, (1967)
- [25] N. K. Karmarkar, *A new polynomial-time algorithm for linear programming*, in : Proceedings of the 16th Annual ACM Symposium on Theory of Computing, 4, 373–395, (1984).

- [26] Z. Kebbiche, *étude et extentions d'algorithmes de points intérieurs pour la programmation non linéaire*, Thèse doctorat d'état, Université Farhat Abbas - Setif, (2007).
- [27] A. Keraghel, *Analyse convexe : théorie fondamentale et exercices*, Editions Dar el'Houda, Ain M'lila, Algérie, (1999).
- [28] A. Keraghel, *Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar*, Thèse de Doctorat, Université Joseph Fourier – Grenoble I, France, (1989).
- [29] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady, 20 191–194, (1979).
- [30] B. Kheirfam, M. Moslem, *A polynomial-time algorithm for linear optimization based on a new kernel function with trigonometric barrier term*, Yugoslav Journal of Operations Research, 25(2), 233–250, (2015).
- [31] V. Klee, G. Minty, *How good is the simplex algorithm ?*, In Shisha, Oved. Inequalities III (Proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, Calif., September 1–9, 1969, dedicated to the memory of Theodore S. Motzkin). New York-London : Academic Press, 159–175, (1972).
- [32] I. J. Lustig, *A pratical approach to Karmarkar's algorithm*, Technical report sol 85-5 Systems optimizatioin laboratory ; dep of operations res. STANFORD. univ ; Stanford California 94305, (1985).
- [33] Xin Li, Mingwang Zhang, *Interior-point algorithm for linear optimization based on a new trigonometric kernel function*, Operations Research Letters, 43(5), 471–475, (2015).
- [34] N. Megiddo, *Pathways to the optimal set in linear programming*, in : N. Megiddo (Ed.), Progress in Mathematical Programming : Interior Point and Related Methods, Springer-Verlag, New York, 131–158, (1989).
- [35] R. Naseri, A.Valinejad, *An extended variant of Karmarkar's interior point algorithm*, Applied Mathematics and computation, 184, 737–742 (2007).

- [36] Y. E. Nesterov, A. S. Nemirovski, *Interior-point polynomial methods in convex programming*, SIAM Studies in Applied Mathematics. SIAM Publications, Philadelphia, (1994).
- [37] M. Padberg, *A different convergence proof of the projective method for linear programming*, New York University, (1985).
- [38] J. Peng, C. Roos, T. Terlaky, *Self-regular functions and new search directions for linear and semidefinite optimization*, Mathematical Programming, 93, 129–171, (2002).
- [39] J. Peng, C. Roos, and T. Terlaky, *A new and efficient large-update interior-point method for linear optimization*, Journal of Computational Technologies, 6, 61–80, (2001).
- [40] J. Peng, C. Roos, T. Terlaky, *Self-Regularity : A New Paradigm for primal-Dual Interior-Point Algorithms*, Princeton University Press, Princeton, NJ, (2002).
- [41] M. R. Peyghami, S. Fathi Hafshejani, *Complexity analysis of an interior-point algorithm for linear optimization based on a new proximity function*, Numerical Algorithms, 67, 33–48, (2014).
- [42] M. R. Peyghami, S. Fathi Hafshejani, L. Shirvani, *Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function*, Journal of Computational and Applied Mathematics, 255, 74–85, (2014).
- [43] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Mathematical Programming, 40 59–93, (1988).
- [44] R. T. Rockafellar, *Convex analysis*, published by Princeton University Press, 41 William Street, Princeton, New Jersey 08540, (1988).
- [45] C. Roos, T. Terlaky, J. Ph. Vial, *Theory and algorithms for linear optimization, An Interior-Point Approach*, John Wiley & Sons, Chichester, UK, (1997).
- [46] C. Roos, T. Terlaky, J. Vial, *Optimization Theory and Algorithm for Linear Programming Optimization*, Princeton University, (2001).

- [47] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, New York, (1986).
- [48] D. F. Shanno, C. Bergchi, *Computig Karmarkar projective quickly*, Matheatical Programming, 41, 61–71, (1988).
- [49] N. Z. Shor, *Utilization of the operation of space dilatation in the minimization of convex functions*, Kibernetika, 1, 6–12, (1970).
- [50] G. Sonnevend, *An “analytic center” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in : A. Prekopa, J. Szelezsan, B. Strazicky (Eds.), *System Modelling and Optimization : Proceedings of the 12th IFIP-Conference*, Budapest, Hungary, 1985, in : *Lecture Notes in Control and Inform. Sci.*, vol. 84, Springer-Verlag, Berlin, 866–876, (1986).
- [51] Latriece Y. Tanksley, *Interior Point Methods and Kernel Functions of a Linear Programming Problem*, Doctoral thesis, Georgia Southern University, États-Unis, (2009).
- [52] M. VC. Vieira, *The Accuracy of Interior-Point Methods Based on Kernel Functions*, Journal of Optimization Theory and Applications, 155, 637–649, (2012).
- [53] S. J. Wright, *primal-dual interior point Methods*, SIAM, Philadelphia, USA, (1997).
- [54] XM. Yang, KL. Teo, XQ. Yang, *A Characterization of Convex Function*, Applied Mathematics Letters, 13, 27–30, (2000).
- [55] A. Yassine, *Etudes adaptatives et comparatives de certains Algorithmes en optimisation*, Thèse de Doctorat, Université Joseph Fourier –Grenoble I, France, (1989).
- [56] Y. Ye, *Interior point algorithms. Theory and Analysis*, John-Wiley. Sons, Chichester, UK, (1997).