



Faculté des Sciences Exacte et Informatique
Département de Mathématiques

N° d'ordre :.....

N° de séries :.....

Mémoire de fin d'études

Présenté pour l'obtention du diplôme de

Master

Spécialité : Mathématiques.

Option : EDP et Applications.

Thème

La Méthode De Point Intérieur Pour La Programmation Linéaire Basée Sur Une Nouvelle Fonction Noyau

Présenté par :

- Kenioua Ilham.
- Menghour Ibtissem.

Devant le jury :

Président	: DJEMAI Samia	MCB	Université de Jijel
Encadreur	: TOUIL Imene	MCB	Université de Jijel
Examineur	: MENNICHE Linda	MCB	Université de Jijel
Examineur	: SOUALHIA Imene	MCB	Université de Jijel

Remerciements

Tout d'abord et avant tout, louage à **Dieu** qui nous a guidé et donné la force, la volonté, la patience et le courage pour accomplir ce modeste travail.

Nous remercions chaleureusement notre encadreur **Mme TOUIL Imene** pour la pertinence de ces remarques et sa patience pendant ce travail. Nous admirons beaucoup ses travaux et sa manière de diriger qui furent pour nous une grande source d'inspiration et de motivation.

Nous adressons également nos vifs remerciements à **Mme DJEMAI Samia** d'avoir présider le jury de contenance.

Nous remercions l'examinatrice **Mme MENNICHE Linda** pour avoir accepté d'être membre de ce jury.

Nous remercions aussi l'examinatrice **Mme SOUALHIA Imene** pour avoir accepté d'être membre de ce jury.

Mes plus vifs remerciements à mes enseignants du département de mathématiques surtout de Ma spécialité **EDP et applications** pour leur sincérité.

Merci à ma famille : *mon père, ma mère, mes sœurs et mes frères.*

Dédicace

Je dédie ce mémoire

*A mes chères parents ma mère et mon père .
Pour leur confiance, leur patience, leur soutien,...*

A mes frères et mes sœurs.

A mon oncle et sa femme.

A toute ma famille et mes chère amies.

A Mohammed.

A toute qui m'aiment et que j'aime.

ilham

Table des matières

Introduction	1
1 Préliminaires et notions fondamentales	3
1.1 Analyse convexe	3
1.1.1 Ensembles et fonctions convexes	3
1.1.2 Ensembles et fonctions affines	4
1.1.3 Convexité et dérivée	5
1.1.4 Fonction barrière	6
1.2 Programme mathématique	6
1.2.1 Définitions	6
1.2.2 Classification d'un programme mathématique	8
1.2.3 Principaux résultats d'existence et d'unicité	8
1.2.4 Conditions d'optimalité	9
1.2.5 Quelques programmes mathématiques	10
1.3 Programmation linéaire	13
1.3.1 Formes usuelles d'un programme linéaire	13
1.3.2 Dualité en programmation linéaire	15
1.3.3 Relation primal-dual pour la programmation linéaire	15

2	Résolution d'un programme linéaire (PL)	17
2.1	Résultats fondamentaux	18
2.2	Méthodes de résolution d'un programme de (PL)	19
2.2.1	Méthode (TC) primale	19
2.2.2	Méthode (TC) duale	22
2.2.3	Méthode (TC) primale-duale	23
3	Méthode de trajectoire centrale via les fonctions noyaux	28
3.1	Méthode (TC) primale-duale basée sur les fonctions noyaux	29
3.1.1	Introduction de la notion des fonctions noyaux.	29
3.1.2	Algorithme prototype de la méthode (TC) primale-duale	31
3.2	Nouvelle fonction noyau	31
3.2.1	Propriétés de la nouvelle fonction noyau	31
3.2.2	Quelques fonctions noyaux et leur complexité	32
3.2.3	Analyse de la complexité	41
	Bibliographie	52

Notations et terminologies

\mathbb{R}^n	:	l'ensemble des vecteurs de n composantes réelles.
\mathbb{R}_+^n	:	l'orthant positif de l'espace \mathbb{R}^n .
\mathbb{R}_{++}^n	:	l'orthant strictement positif de l'espace \mathbb{R}^n .
S^n	:	l'ensemble des matrices symétriques d'ordre n .
S_{++}^n	:	l'ensemble des matrices symétriques définies positives .
x^T	:	le transposé du vecteur x de \mathbb{R}^n .
$\ \cdot\ $:	la norme 2.
e	:	le vecteur n -dimensionnel des uns.
x_1	:	la plus petite composante de vecteur x .
xs	=	$(x_1s_1, \dots, x_ns_n)^T$ (produit de Hadamard).
X	=	$diag(x)$.
S	=	$diag(s)$.
x^{-1}	=	$(\frac{1}{x_1}, \dots, \frac{1}{x_n})$, ($x_i \neq 0$).
X^{-1}	=	$diag(x^{-1})$.
$\Delta x, \Delta s$:	les directions de Newton.
S_{int}	=	$\{x \in \mathbb{R}_{++}^n : Ax = b\}$ l'ensemble des solutions strictement réalisables de (P).
T_{int}	=	$\{(y, s) \in \mathbb{R}^m \times \mathbb{R}_{++}^n : A^T y + s = c\}$ l'ensemble des solutions strictement réalisables de (D).
W	=	$\{(x, y, s) \in \mathbb{R}_{++}^n \times \mathbb{R}^m \times \mathbb{R}_{++}^n : Ax = b, A^T y + s = c\}$ l'ensemble des solutions strictement réalisables de (P) et (D).
$f(x)$ et $g(x)$:	fonctions de \mathbb{R}_{++}^n dans \mathbb{R}_{++}^n .
$f(x)=O(g(x))$	=	$f(x) \leq C_1g(x)$ pour certains constante positive C_1 .
$f(x) = \Theta(g(x))$	=	$C_2g(x) \leq f(x) \leq C_3g(x)$ pour certains constante positive C_2 et C_3 .

Introduction

La programmation linéaire (*PL*) constitue l'une des acquisitions plus importantes de la théorie économique d'après la deuxième guerre mondiale. Elle s'est développée très rapidement, grâce aux efforts conjugués des mathématiciens, des chefs d'entreprises, des chefs militaires, des statisticiens et des économistes. Ceci nous a permis de la considérer comme l'un des plus beaux succès de la recherche opérationnelle. Un programme linéaire est un problème qui consiste à optimiser (maximiser ou minimiser) une fonction linéaire à plusieurs variables soumise à un ensemble de contraintes linéaires. La méthode du simplexe a été considérée comme la meilleure méthode de résolution jusqu'aux années 70, lorsque Klee et Minty (1972) ont trouvé pour la première fois un exemple qui montrait que la méthode du simplexe pouvait prendre un nombre d'itérations exponentiel (i.e., l'algorithme de la méthode a une complexité exponentielle). A cet égard, la recherche d'un algorithme polynomial pour la programmation linéaire était lancée. La réponse est venue de Khachiyan en (1979) lorsqu'il a montré que la méthode des ellipsoïdes (un algorithme de points intérieurs développé au début des années 70 pour la programmation non linéaire) était de complexité polynomiale, lorsqu'elle est appliquée à la programmation linéaire, mais cette dernière ne fonctionnait pas bien en pratique. En 1984, Karmarkar [6] a proposé un algorithme de complexité polynomiale efficace en pratique, basé sur une méthode de points intérieurs. On désigne par méthode de points intérieurs (*MPIs*), toute procédure de résolution générant une suite de points appartenant à l'intérieur (relatif) du domaine réalisable (admissible) et convergeant vers la solution optimale du programme considéré. Den Hertog (1994) [7] a classé les méthodes points intérieurs en trois catégories : les méthodes affines, les méthodes de réduction du potentiel et les méthodes de trajectoire centrale. Dans ce travail, on s'intéresse plus particulièrement, aux méthodes de type trajectoire centrale (*TC*) primale-duale, à cause de ses avantages par rapport aux autres classes de méthodes de points intérieurs. La plupart de ces méthodes sont basées sur la fonction barrière logarithmique. En 2001, Peng et autres [8] ont proposé une nouvelle fonction noyau non logarithmique, telles que cette fonction soit convexe et coercive sur

son domaine réalisable, dans le but d'améliorer la complexité algorithmique de *MPIs*. En 2004, Bai et autres ont introduit pour la première fois une fonction noyau à terme barrière trigonométrique. En 2012, El ghami et autres[5] ont évalué la complexité algorithmique de la fonction noyau proposée par Bai et autres. Les travaux se poursuivent à présent, on trouve ainsi ceux de Peyghami et autres [9, 10] en 2014, qui ont proposé deux différentes fonctions noyaux à termes trigonométrique-logarithmique et trigonométrique-exponentiel. La même année, Cai et autres [3] ont proposé une autre fonction noyau à terme barrière trigonométrique-logarithmique pour les méthodes de points intérieurs de type primal-dual. En 2016, Bouafia et autres [2] ont proposé une fonction noyau généralisée avec une fonction barrière de type trigonométrique, ils ont prouvé théoriquement et numériquement que leur complexité algorithmique est la meilleure pour les méthodes de points intérieurs à grand-pas basées sur les fonctions noyaux à terme trigonométrique. Pour cette raison, on a choisi de détailler leur article intitulé : « An Efficient Primal-dual Interior Point Method For Linear Programming Based On A New Kernel Function With A Trigonometric Barrier Term ».

Ce mémoire est organisé comme suit

Le premier chapitre présente un rappel des notions fondamentales d'usage fréquent pour la suite, à savoir : l'analyse convexe, la programmation mathématique (*PM*) et les principaux résultats d'existence et d'unicité d'une solution optimale d'un *PM*. Une présentation générale des différentes versions de méthodes de *TC* réalisables appliquées à la programmation linéaire (*PL*) est proposée dans le deuxième chapitre. Dans le troisième chapitre, on a présenté d'une manière détaillée le papier publié par Bouafia et autres en 2016.

Chapitre 1

Préliminaires et notions fondamentales

Dans ce chapitre, nous rappelons quelques notions de base sur l'analyse convexe, la programmation mathématique et quelques différents modèles d'un programme mathématique citons par exemple la programmation linéaire (PL), la programmation non linéaire (PNL) et la programmation quadratique (PQ). On termine ce chapitre par la présentation de (PL), où on donne les différentes formes de (PL) et la notion de dualité en (PL), ainsi que les principaux résultats d'existence et d'unicité.

1.1 Analyse convexe

1.1.1 Ensembles et fonctions convexes

Dans cette section, on présente un rappel des notions fondamentales de l'analyse convexe qui serviront d'appuis pour la suite.

Définition 1.1.

1. Un ensemble C de \mathbb{R}^n est dit convexe si

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

2. Un ensemble C est un polyèdre convexe s'il s'écrit comme suit

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

Où A_i est un vecteur non nul de \mathbb{R} et b_i est un scalaire pour $i = \overline{1, m}$.

3. Une fonction f est convexe sur un ensemble convexe C si

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

4. Une fonction f est strictement convexe sur un ensemble convexe C si

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

5. On dit que f est (strictement) concave sur un ensemble convexe si $(-f)$ est (strictement) convexe sur C .

Définition 1.2. Un sous ensemble D est un cône si et seulement si

$$\forall x \in D, \lambda \geq 0 : \lambda x \in D.$$

Si D est convexe, on dit que D est un cône convexe.

1.1.2 Ensembles et fonctions affines

Définition 1.3. Un sous ensemble F de \mathbb{R}^n est dite affine si

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

Définition 1.4. Étant donné $S \subseteq \mathbb{R}^n$, alors il existe une partie affine unique $F \subseteq \mathbb{R}^n$ contenant S appelée enveloppe affine de S et notée $\text{aff}(S)$. Autrement dit $\text{aff}(S)$ est la plus petite partie affine de \mathbb{R}^n contenant S .

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

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

Définition 1.5. Une fonction f définie de \mathbb{R}^n dans \mathbb{R}^m est dite affine si

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

1.1.3 Convexité et dérivée

Définition 1.6. Soit $f : U \subset E \rightarrow F \subset \mathbb{R}$, avec U un ouvert de $E \subset \mathbb{R}^n$ et a un point de U , on dit que f est une fonction différentiable en a s'il existe une application linéaire $L \in \mathcal{L}(E, F)$ telle que

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

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

Définition 1.7. • On dit que f est deux fois différentiable en a si f est différentiable sur un voisinage de a et si df est elle-même différentiable en a .

- On dit que f est deux fois différentiable sur U si f est deux fois différentiable en tout point de U .

Définition 1.8. Soit f une fonction différentiable en tout point de $U \subset E \subset \mathbb{R}^n$, on peut considérer l'application

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

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

Définition 1.9. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continûment différentiable, leur gradient au point $x \in \mathbb{R}^n$ s'écrit

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

Définition 1.10. Soit U un ouvert de \mathbb{R}^n et $f : U \rightarrow \mathbb{R}$ une fonction deux fois différentiable en $a \in U$ dont toutes les dérivées partielles d'ordre deux sont définies en a , alors la matrice

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

est appelée matrice Hessienne de f en a .

Définition 1.11. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois continûment différentiable sur un domaine convexe C .

1. On dit que f est une fonction convexe sur C si et seulement si la matrice Hessienne est semi-définie positive, i.e.,

$$\forall x \in C, y^T \nabla^2 f(x) y \geq 0, \forall y \in \mathbb{R}^n.$$

2. On dit que f est une fonction strictement convexe sur C si et seulement si la matrice Hessienne est définie positive, i.e.,

$$\forall x \in C, y^T \nabla^2 f(x) y > 0, \forall y \in \mathbb{R}^n - \{0\}.$$

Définition 1.12. Une fonction f est dite coercive sur un ensemble convexe C si

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

1.1.4 Fonction barrière

Définition 1.13. On appelle fonction barrière toute fonction f qui vérifie

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

Remarque. Dans la programmation mathématique, la fonction barrière classique la plus utilisée est la fonction barrière logarithmique.

1.2 Programme mathématique

1.2.1 Définitions

Dans cette partie, on donne les outils de base d'un problème d'optimisation. On rappelle certaines définitions élémentaires, les principaux résultats d'existence et d'unicité de la solution optimale et les conditions d'optimalité.

Problème d'optimisation

- Un problème d'optimisation sans contraintes s'écrit comme suit

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

Autrement dit

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

- Un problème d'optimisation avec contraintes (ou programme mathématique) s'écrit sous la forme

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

Autrement dit

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

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

Programmation mathématique

La programmation mathématique constitue un domaine vaste et riche dans l'analyse numérique. Elle traite plusieurs modèles mathématiques et problèmes pratiques importants. D'une façon générale, un programme mathématique est un problème d'optimisation avec contraintes de type

$$(PM) \quad \begin{cases} \min (\text{ou } \max) f(x) \\ S = \{x \in \Omega \subseteq \mathbb{R}^n : g_i(x) \leq 0, i = \overline{1, k} \text{ et } h_j(x) = 0, j = \overline{1, m}, k + m = n\}. \end{cases}$$

Où f , g_i et h_j sont des fonctions définies de \mathbb{R}^n dans \mathbb{R} .

- La fonction f est appelée fonction objectif ou économique.
- Un point x de S est appelé solution réalisable de (PM) .
- L'ensemble S est appelé l'ensemble des solutions réalisables de (PM) ou l'ensemble des contraintes ou tout simplement "le domaine réalisable".

Définition 1.14. (*Minimum local*)

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

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

L'ensemble des minimas locaux de (PM) et noté par : $\underset{S}{loc \min} f(x)$.

Définition 1.15. (*Minimum global*)

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

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

L'ensemble des minimas globaux de (PM) et noté par $\arg \min_S f(x)$.

1.2.2 Classification d'un programme mathématique

On classe le problème (PM) à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité des fonctions du problème. En effet, (PM) est dit convexe si les fonctions f et g_i sont convexes et les fonctions h_j sont affines. Si les fonctions f , g_i et h_j sont toutes différentiables on dit que (PM) est différentiable. La classe des (PM) convexes différentiables est le cas le mieux élaboré, les programmes non convexes ou non différentiables sont difficiles à traiter.

1.2.3 Principaux résultats d'existence et d'unicité

Théorème 1.16. (*Weierstrass*)

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

Corollaire 1.17. Soit S un ensemble non vide et fermé de \mathbb{R}^n , f est une fonction continue et coercive sur S , alors (PM) admet au moins une solution optimale globale $x^* \in S$.

Théorème 1.18. Si S est convexe et f est strictement convexe alors, il existe au plus une solution optimale de (PM) .

Remarque. L'unicité d'une éventuelle solution optimale est en souvent une conséquence de la stricte convexité de la fonction objectif f et de la convexité du domaine réalisable de (PM) .

1.2.4 Conditions d'optimalité

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

Qualification des contraintes

1. Si S est un polyèdre convexe (i.e., g_i et h_j affines), alors par définition les contraintes sont qualifiées en tout point réalisable.
2. Si S est convexe (i.e., g_i convexes et h_j affines) et $\text{int}S \neq \emptyset$, alors les contraintes sont qualifiées partout. C'est la condition de Slater.
3. Une contrainte d'inégalité $g_i(x) \leq 0$ est dite saturée (ou active) en $x^* \in S$ si $g_i(x^*) = 0$. De ce fait, une contrainte d'égalité $h_j(x) = 0$ est par définition saturée en tout point $x \in S$. Les contraintes sont qualifiées en $x^* \in S$ si les gradients de toutes les fonctions contraintes saturées en x^* sont linéairement indépendants.

Le lagrangien d'un programme mathématique (PM) est défini par

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

Théorème 1.19. (Karush Kuhn Tucker **KKT**)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable sur S , si x^* est un minimum local du problème (PM) , alors il existe un vecteur $y \in \mathbb{R}^m$ et $\lambda \in \mathbb{R}_+^k$ tel que

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^k \lambda_i \nabla g_i(x^*) + \sum_{j=1}^m y_j \nabla h_j(x^*) = 0, & (\text{Condition d'optimalité}). \\ \lambda_i g_i(x^*) = 0, i = 1, \dots, k, & (\text{Condition de complémentarité}). \\ h_j(x^*) = 0, j = 1, \dots, m. \end{cases}$$

Définition 1.20. On dit qu'un problème de programmation mathématique est convexe s'il consiste à minimiser une fonction convexe (ou maximiser une fonction concave) sur un domaine convexe.

Ainsi le problème (PM) est un problème de programmation convexe (ou simplement un programme convexe) si f est convexe, les fonctions g_i ($i = 1, \dots, k$) et h_j ($j = 1, \dots, m$) sont convexes et $\Omega \subset \mathbb{R}^n$ est aussi convexe.

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

La propriété fondamentale des programmes convexes apparaît alors dans le résultat suivant

Théorème 1.21. *Pour un programme convexe, tout optimum local est un optimum global.*

1.2.5 Quelques programmes mathématiques

Rappelons qu'un programme mathématique est un problème d'optimisation sous contraintes. Il existe plusieurs types de programme mathématique, parmi les cas les plus étudiés on trouve

Programme linéaire

Un problème de programmation linéaire est un problème de programmation mathématique convexe dans lequel la fonction objectif et les contraintes sont linéaires. Les spécialistes du domaine, le considèrent comme étant la technique la plus utilisée dans la recherche opérationnelle.

Définition 1.22. *Un programme linéaire (PL) est un système d'équations ou d'inéquations appelées "contraintes" qui sont linéaires et à partir de ses contraintes on doit optimiser une fonction également linéaire appelée "objectif" (économique) sous la forme*

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

(PL) s'écrit aussi sous la forme matricielle

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

Où (PL) est considéré comme primal.

Tels que

- $\text{opt} = \max$ ou \min .
- $c^t x = \sum_{j=1}^n c_j x_j$.

- $c \in \mathbb{R}^n$ est le vecteur coût.
- $b \in \mathbb{R}^m$ est le second membre.
- $a_{ij} \in \mathbb{R}, \forall i = 1, \dots, m, \forall j = 1, \dots, n$ est la matrice des contraintes.

Exemple 1.23.

$$(PL) \quad \begin{cases} \max (5x_1 + 10x_2) \\ x_1 + x_2 \leq 100, \\ x_1 + x_2 \geq 200, \\ x_1, x_2 \geq 0. \end{cases}$$

Programme quadratique

La programmation quadratique est connue par ses applications multiples dans plusieurs domaines. Sans perte de généralité, on peut présenter un programme quadratique sous la forme suivante

$$(PQ) \quad \begin{cases} \min \frac{1}{2}x^T Qx + c^T x \\ Ax = b, \\ x \geq 0, \end{cases}$$

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

Autrement dit, un programme quadratique est un programme mathématique où sa fonction objectif est quadratique et ses contraintes sont linéaires.

Remarque. • *L'ensemble des contraintes S de (PQ) est un polyèdre convexe et fermé, la fonction objectif est infiniment différentiable.*

• *(PQ) est convexe si et seulement si f est convexe auquel cas la matrice Q est semi définie positive.*

Exemple 1.24.

$$(PQ) \quad \begin{cases} \max (5x_1 + 10x_2 + \frac{1}{2}(x_1^2 + 4x_1x_2 + x_2^2)) \\ x_1 + x_2 = 100, \\ 3x_1 + x_2 = 200, \\ x_1, x_2 \geq 0. \end{cases}$$

Où

$$A = \begin{pmatrix} 1 & 1 \\ 3 & 1 \end{pmatrix} \text{ et } Q = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

Programme non linéaire

La programmation non linéaire est la recherche de l'optimum d'une fonction non linéaire sur un sous ensemble convexe ou non d'un espace donné. Le problème non linéaire s'écrit sous la forme

$$(PNL) \quad \begin{cases} \max f(x) \\ g_i(x) \leq 0, i = 1, \dots, n, \\ h_j(x) = 0, j = 1, \dots, m, \\ x \in \mathbb{R}^n. \end{cases}$$

Donc, un programme mathématique est dit non linéaire si f est non linéaire et S est un domaine convexe ou non convexe.

Ainsi la programmation non linéaire se présente comme étant une généralisation de la programmation quadratique et de la programmation linéaire.

Exemple 1.25.

$$(PNL) \quad \begin{cases} \min(x_1^3 + x_2) \\ x_1^3 x_2 \geq 1, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Programme semi-définie

La programmation semi-définie (*SDP*) est une généralisation de la programmation linéaire (*P*), où les vecteurs sont remplacés par des matrices et l'orthant positif \mathbb{R}_+^n par le cône convexe S_+^n .

Le problème primal de la programmation semi-définie sous forme standard, s'écrit comme suit

$$(SDP) \quad \begin{cases} \min \langle C, X \rangle \\ \text{tr}(A_i) = b_i, i = 1, \dots, m, \\ X \in S_+^n \end{cases}$$

C et $A_i, i = 1, \dots, n$ sont des matrices, $\langle C, X \rangle = \text{tr}(C^t X) = \sum_{i,j=1}^m C_{ij} X_{ij}$ et $b \in \mathbb{R}^n$.

Exemple 1.26.

$$(SDP) \quad \begin{cases} \min \langle C, X \rangle \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, 4, \\ X \in \mathbb{S}_+^3. \end{cases} \quad (1.1)$$

Où

$$C = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Remarque. Si X est une matrice diagonale, le problème (SDP) se réduit à un problème de programmation linéaire.

Il existe d'autres modèles de programmation mathématique tel que : Les programmes convexes où f et g_i sont convexes et les h_j sont affines, Les programmes en nombres entiers où l'ensemble des variables est un ensemble discret, c'est-à-dire les variables à valeurs entières.

1.3 Programmation linéaire

1.3.1 Formes usuelles d'un programme linéaire

Les programmes linéaires (PL) se présentent sous trois formes différentes sont

Forme standard

Un programme linéaire (PL) est sous forme standard si toutes les contraintes sont des égalités, c'est-à-dire qu'il s'écrit sous la forme

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

Forme canonique

Un programme linéaire (PL) est sous forme canonique si toutes les contraintes sont des inégalités, c'est-à-dire qu'il s'écrit sous la forme

$$(PL) \quad \begin{cases} \text{opt}(c^T x) \\ Ax (\leq \text{ou} \geq) b, \\ x \geq 0. \end{cases}$$

Forme mixte

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

Remarque. *On peut toujours mettre un programme linéaire quelconque sous forme standard en introduisant des variables supplémentaires positives appelées "variables d'écart" ou "variables artificielles".*

Exemple 1.27.

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

En introduisant les variables d'écart $x_3 \geq 0$, $x_4 \geq 0$ dans la première et la deuxième contrainte, le problème précédent s'écrit sous forme standard comme suit

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

1.3.2 Dualité en programmation linéaire

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

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

Son dual s'écrit sous la forme

$$(D) \quad \begin{cases} \max(b^T y) \\ A^T y \leq c, \\ y \in \mathbb{R}^m. \end{cases}$$

Exemple 1.28. Soit le programme primal sous forme standard

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

Son dual est

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

1.3.3 Relation primal-dual pour la programmation linéaire

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

Primal	min	max	Dual
Contraintes	$\geq b_i$	≥ 0	Variables
	$\leq b_i$	≤ 0	
	$= b_i$	<i>libre</i>	
Variables	≥ 0	$\geq c_j$	Contraintes
	≤ 0	$\leq c_j$	
	<i>libre</i>	$= c_j$	

1. Le nombre de variables duales est égal au nombre de contraintes primales, de même le nombre de contraintes duales est égal au nombre de variables primales.
2. Le dual du problème dual (D) est le problème primal (P).
3. On peut transformer un problème de maximisation à un problème de minimisation, il suffit d'écrire $\max(f) = -\min(-f)$ (respectivement $\min(f) = -\max(-f)$).

Remarque. Dans notre étude, on s'intéresse à la résolution des problèmes de programmation linéaire. La résolution complète de (PL) traite dans l'ordre les points suivants

1. L'existence (et éventuellement l'unicité) d'une solution optimale.
2. La caractérisation de la solution (il s'agit des conditions d'optimalité).
3. L'élaboration d'algorithmes pour calculer cette solution.

Chapitre 2

Résolution d'un programme linéaire (PL)

Dans ce chapitre, on s'intéresse à la résolution des problèmes de programmation linéaire sous forme standard de type

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

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

Bien évidemment, n'importe quel (PL) se ramène facilement à cette forme.

Le dual de (P), noté (D) est donné par

$$(D) \quad \begin{cases} \max b^T y \\ A^T y + s = c, \\ s \geq 0, y \in \mathbb{R}^m \end{cases}$$

Posons une fois pour toutes les hypothèses suivantes

(H1) : S_{int} est non vide,

(H2) : T_{int} est non vide,

(H3) : A est de plein rang ($rg(A) = m < n$),

où

$$S_{int} = \{x \in \mathbb{R}_+^n : Ax = b, x > 0\},$$

$$T_{int} = \{(y, s) \in \mathbb{R}^m \times \mathbb{R}_+^n : A^T y + s = c, \}.$$

Ces hypothèses sont nécessaires pour l'existence des solutions réalisables.

Proposition 2.1. *Un programme linéaire réalisable et borné (objectif borné) possède au moins une solution optimale, situé sur la frontière du domaine réalisable.*

2.1 Résultats fondamentaux

Théorème 2.2. *(dualité faible)*

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

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

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

■

Théorème 2.3. *(dualité forte)*

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

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

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

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

$$b^T y^* = c^T x^* \geq \min c^T x \geq \max b^T y \geq b^T y^* = c^T x^*,$$

et comme $b^T y^* = c^T x^*$, alors

$$\left\{ \begin{array}{l} c^T x^* \geq \min c^T x \geq c^T x^*, \\ \text{et} \\ b^T y^* \leq \max b^T y \geq b^T y^*. \end{array} \right.$$

D'où le résultat. ■

Théorème 2.4. *(Théorème de dualité)*

- *Si l'un des problèmes (P) et (D) admet une solution optimale fini, il en est de même pour l'autre et leur valeurs optimales correspondantes sont égales.*
- *Si l'un des problèmes à une valeur optimale non borné, l'autre n'a pas de solution optimale.*

Théorème 2.5. (Théorème des écarts complémentaire)

Soient \bar{x} et \bar{y} deux solutions réalisables de (P) et (D) respectivement, posons $\bar{s} = c - A^T \bar{y}$ vecteur des variables d'écart associé à \bar{y} , alors \bar{x} et \bar{s} sont optimales si et seulement si $\bar{x}_i \bar{s}_i = 0, \forall i = 1, \dots, n$.

2.2 Méthodes de résolution d'un programme de (PL)

On dispose de deux types de méthodes de résolution d'un programme linéaire, méthode simpliciale à complexité exponentielle et méthodes de points intérieurs qui s'avèrent actuellement efficaces pour les problèmes de grandes tailles à cause de leur complexité polynomiale.

Les différents types de méthodes de point intérieurs sont présentés dans la littérature en trois classes principales

- Les méthode affines.
- Les méthodes projectives.
- Les méthodes de trajectoire centrale (TC).

Principe des méthodes (TC)

L'idée générale de ces méthodes consiste à suivre un chemin particulier (chemin des centres) constitué d'une suite de points intérieurs en prenant comme direction de déplacement celle de Newton et un pas de déplacement convenable choisi d'une façon à maintenir la stricte faisabilité des nouveaux itérés.

On distingue trois variantes des méthodes de (TC) à s'avoir

- méthode primale.
- méthode duale.
- méthode primale-duale.

2.2.1 Méthode (TC) primale

Reprenons le programme linéaire standard (P)

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

La plus part des méthodes nouvelles de points intérieurs peuvent être vues comme des variantes des méthodes barrières logarithmique de Frisch (1955).

En effet, au problème (P) on associe le problème non linéaire suivant

$$(P_\mu) \quad \begin{cases} \min c^T x - \mu \sum_{i=1}^n \ln x_i = f_\mu(x) \\ Ax = b, \\ x > 0. \end{cases}$$

L'objectif f_μ de (P_μ) est appelé fonction barrière logarithmique et $\mu > 0$ désigne le paramètre barrière.

La résolution de (P_μ) est équivalente à celle de (P) au sens que si x_μ^* est une solution optimale de (P_μ) , alors $x^* = \lim x^*(\mu)$ (quand $\mu \rightarrow 0$) est une solution optimale de (P). Il suffit donc de résoudre (P_μ) .

Proposition 2.6. *Sous l'hypothèse (H1), on a*

Pour tout $\mu > 0$, l'ensemble des solutions optimales de (P) est non vide et borné.

Démonstration. Les contraintes de (P_μ) sont linéaires et l'objectif f_μ est une fonction strictement convexe puisque $f''(x) = \mu X^{-2}$ est définie positive donc, si la solution existe elle est unique, globale et complètement caractérisée par les conditions de **KKT**, i.e., x optimale si et seulement si $\exists y \in \mathbb{R}^m$ tel que

$$\begin{cases} c - \mu X^{-1}e - A^T y = 0, \\ Ax = b, \\ x > 0. \end{cases} \quad (2.1)$$

Le saut de dualité est définie par

$$c^T x - b^T y = \mu x^T X^{-1} e = n\mu,$$

d'où si μ tend vers zéro, $c^T x$ et $b^T y$ convergent vers la même valeur optimale. Par conséquent x et y sont des solutions optimales de (P) et (D) respectivement (d'après le Théorème ??). ■

Description technique

Conformément au principe des méthodes (TC) réalisables, le système non linéaire (2.1) sera résolu par la méthode de Newton, ceci peut être justifié par les résultats magnifiques

obtenus éventuellement en théorie et par les subtilités numériques offertes par cette méthode.

Ainsi, on se limite à chercher des solutions approximatives suivant la trajectoire centrale en formant une suite décroissante $\left\{ \mu^k = \frac{(x^k)^T s^k}{n} \right\}$ convergeant vers zéro.

Plus précisément, il s'agit de résoudre le système $F(x, y) = 0$ tel que

$$F(x, y) = \begin{pmatrix} c - \mu X^{-1}e - A^T y \\ Ax - b \end{pmatrix}$$

D'où $x_+ = x + \alpha \Delta x$, $y_+ = y + \Delta y$, avec $\alpha > 0$ est le pas de déplacement introduit de façon à maintenir $x_+ > 0$ et $\Delta w = (\Delta x, \Delta y)$ est solution du système linéaire

$$DF(x, y)\Delta w = -F(x, y). \quad (2.2)$$

avec $DF(x, y)$ est la matrice jacobienne de F au point (x, y) donnée par

$$DF(x, y) = \begin{pmatrix} \mu X^{-2} & -A^T \\ A & 0 \end{pmatrix}$$

D'après (2.2), on obtient le système

$$\begin{pmatrix} \mu X^{-2} & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -(c - \mu X^{-1}e - A^T y) \\ 0 \end{pmatrix}$$

On change μ ($\hat{\mu} = \sigma \mu$, $0 < \sigma < 1$) et on réitère jusqu'à l'optimalité.

L'algorithme correspondant à cette version peut être écrit comme suit

Algorithme de la méthode (TC) primale

Dans cette partie, on présente l'algorithme prototype de la méthode de (TC) primale.

Début algorithme

- Trouver $X^0 \in S_{int}$, $y^0 \in \mathbb{R}^m$, $\mu^0 > 0$, $k = 0$;
- Tant que le test d'optimalité n'est pas satisfait ($c^T x - b^T y > \epsilon$) faire

1. Résoudre le système linéaire suivant

$$\begin{pmatrix} \mu^k (X^k)^{-2} & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} -(c - \mu^k (X^k)^{-1}e - A^T y^k) \\ 0 \end{pmatrix};$$

2. Trouver $0 < \alpha \leq 1$ / $x^{k+1} = x^k + \alpha \Delta x^k > 0$;
3. $y^{k+1} = y^k + \Delta y^k > 0$;

$$4. \mu^{k+1} = \sigma \mu^k, \quad 0 < \sigma < 1;$$

$$5. k = k + 1;$$

Fin tant que

Fin Algorithme

2.2.2 Méthode (TC) duale

Considérons le programme linéaire standard (D)

$$(D) \quad \begin{cases} \max b^T y \\ A^T y + s = c, \\ s \geq 0, \end{cases}$$

auquel, on associe le problème barrière suivant

$$(D_\mu) \quad \begin{cases} \max b^T y - \mu \sum_{i=1}^n \ln s_i \\ A^T y + s = c, \\ s > 0, \end{cases}$$

où $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

Proposition 2.7. *Sous l'hypothèse (H2), on a*

pour tout $\mu > 0$, le problème (D_μ) admet une solution optimale si et seulement si l'ensemble des solutions optimales de (D) est non vide et borné.

Description technique

En appliquant les mêmes techniques utilisées dans la version primale, on obtient le système non linéaire suivant

$$\begin{cases} A^T y + s - c = 0, \\ Ax - b = 0, \\ \mu S^{-1} e - x = 0, \\ s > 0, x > 0. \end{cases} \quad (2.3)$$

On le résout par la méthode de Newton à partir d'un point $x \in \mathbb{R}_{++}^n$ et $(y, s) \in T_{int}$. le nouvel itéré est donné par

$$x_+ = x + \alpha \Delta x, \quad y_+ = y + \Delta y, \quad s_+ = s + \alpha \Delta s,$$

où $\alpha > 0$ est le pas de déplacement introduit de façon à maintenir la stricte positivité de x_+ et s_+ (i.e., $x_+ = x + \alpha\Delta x > 0$ et $s_+ = s + \alpha\Delta s > 0$).

Algorithme de la méthode (TC) duale

L'algorithmique prototype de cette version s'écrit comme suit

Début d'algorithme

- Choisir $x^0 \in \mathbb{R}_{++}^n$, $\mu_0 > 0$, trouver $(y^0, s^0) \in T_{int}$, $k = 0$;
- Tant que le test d'optimalité n'est pas satisfait faire
 1. Résoudre le système linéaire suivant

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ -I & 0 & \mu(S^k)^{-2} \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} -Ax^k + b \\ 0 \\ -\mu(S^k)^{-1}e + x^k \end{pmatrix};$$

2. Trouver $0 < \alpha \leq 1 / x^{k+1} = x^k + \alpha\Delta x^k > 0$ et $s^{k+1} = s^k + \alpha\Delta s^k > 0$;
3. $x^{k+1} = x^+$, $s^{k+1} = s^+$, $y^{k+1} = y^k + \alpha\Delta y^k$;
4. $\mu^{k+1} = \sigma\mu^k$, $0 < \sigma < 1$;
5. $k = k + 1$;

Fin tant que

Fin Algorithme

2.2.3 Méthode (TC) primale-duale

Les méthodes de trajectoire centrale primale-duale ont été introduites au début des années 90. Elles ont attiré une grande attention de la part des chercheurs dans le monde entier et elles montrent en général un excellent comportement pratique et théorique (une complexité polynomiale et une convergence super linéaire).

Description technique

Les méthodes (TC) primales-duales sont basées sur la résolution du système non linéaire suivant

$$\begin{cases} A^T y + s - c = 0, \\ Ax - b = 0, \\ XSe - \mu e = 0, \\ s > 0, x > 0, \end{cases} \quad (2.4)$$

qui peut être obtenu en combinant le système non linéaire primal (2.1) et le système non linéaire dual (2.3), le système admet une solution unique si et seulement si les hypothèses (H1) et (H2) sont vérifiées simultanément.

Visiblement, le système (2.4) est plus convenable au traitement numérique que les systèmes (2.1) et (2.3) (il est moins non linéaire). Donc la version primale-duale est plus intéressante que les deux autres versions précédentes pour les raisons suivantes

- Les résultats théorique sont plus consistants
- Les algorithmes sont plus performants en pratique.

Maintenant, on va résoudre le système (2.4) par la méthode de Newton, c'est-à-dire pour chaque $\mu > 0$, on résout le système linéaire suivant

$$DF(x, y, s)\Delta w = -F(x, y, s),$$

où

$$F(x, y, z) = \begin{pmatrix} A^T y + s - c \\ Ax - b \\ Xs - \hat{\mu}e \end{pmatrix}, \quad \hat{\mu} = \sigma\mu \quad (0 < \sigma < 1).$$

$DF(x, y, s)$ désigne la matrice jacobienne de F au point (x, y, s) et $\Delta w = (\Delta x, \Delta y, \Delta s)$ désigne la direction de Newton.

Par des simples calculs, on arrive à résoudre le système linéaire suivant

$$\begin{cases} S\Delta x + X\Delta s = Xs - \hat{\mu}e, \\ A\Delta x = 0, \\ A^T\Delta y + \Delta s = 0. \end{cases}$$

dont sa solution est

$$\Delta x = [S^{-1} - S^{-1}XA^T(AS^{-1}XA^T)AS^{-1}](Xs - \hat{\mu}e),$$

$$\Delta y = -[(AS^{-1}XA^T)^{-1}AS^{-1}](Xs - \hat{\mu}e),$$

$$\Delta s = -A^T \Delta y.$$

Le nouvel itéré s'écrit sous la forme

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

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

On sait que le nouvel itéré doit être positif, i.e.,

$$x^+ = x + \alpha \Delta x > 0, \quad (2.5)$$

$$s^+ = s + \alpha \Delta s > 0, \quad (2.6)$$

de (2.5), on a

$$\alpha \Delta x > -x.$$

Ce qui est équivalent à

$$\alpha = \alpha_1 = \begin{cases} \min\left(\frac{-x_i}{(\Delta x)_i}\right), & \text{si } I_0 \neq \emptyset, \\ +\infty, & \text{si } I_0 = \emptyset, \end{cases}$$

tel que

$$I_0 = \{i ; \Delta x_i < 0, i = \overline{1, n}\}.$$

De même pour (2.6), on a

$$\alpha \Delta s > -s.$$

Ce qui est équivalent à

$$\alpha = \alpha_2 = \begin{cases} \min\left(\frac{-s_i}{(\Delta s)_i}\right), & \text{si } I_1 \neq \emptyset, \\ +\infty, & \text{si } I_1 = \emptyset, \end{cases}$$

tel que

$$I_1 = \{i ; \Delta s_i < 0, i = \overline{1, n}\}.$$

Alors, il suffit de prendre le pas de déplacement suivant

$$\alpha = \rho \min(\alpha_1, \alpha_2), 0 < \rho < 1.$$

Facteur de centralité

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

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

$$S_{cent}(\sigma) = \{(x, y, s) \in S_{int} \times T_{int}; \|XSe - \mu e\| \leq \sigma\mu; \mu > 0\}$$

Hypothèses : Soient δ et σ deux constantes telles que

$$0 \leq \sigma < \frac{1}{2} \text{ et } 0 < \delta < \sqrt{n}, \quad (2.7)$$

$$\frac{\sigma^2 \delta^2}{2(1-\sigma)} \leq \sigma \left(1 - \frac{\delta}{\sqrt{n}}\right). \quad (2.8)$$

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

Algorithme de la méthode (TC) primale-duale

Dans cette partie, on présente l'algorithme de la méthode (TC) primale-duale, puis on donne quelques résultats qui servent pour l'étude de la convergence et la complexité arithmétique pour cette version.

Début d'algorithme

Données soient $\epsilon > 0$ un paramètre de précision, σ et δ deux constantes satisfaisant (2.7) et (2.8),

Initialisation $(x^0, y^0, s^0) \in S_{cent}(\sigma)$ où $\mu_0 = \frac{(x^0)^T s^0}{n}$ (difficulté majeure), $k=0$;

Tant que $(x^k)^T s^k \geq \epsilon$ faire

1. Prendre $\mu_{k+1} = \mu_k \left(1 - \frac{\delta}{\sqrt{n}}\right)$;

2. calculer

$$\begin{cases} \Delta x = [(S^k)^{-1} - (S^k)^{-1} X^k A^T (A(S^k)^{-1} X^k A^T) A S^{-1}] (X^k S^k e - \mu_{k+1} e); \\ \Delta y = -[(A(S^k)^{-1} X^k A^T)^{-1} A (S^k)^{-1}] (X^k S^k e - \mu_{k+1} e), \\ \Delta s = -A^T \Delta y^k; \end{cases}$$

3. Prendre $\alpha = \rho \min(\alpha_1^k, \alpha_2^k)$, $0 < \rho < 1$;

4. Poser

$$x^{k+1} = x^k + \alpha^k \Delta x^k;$$

$$y^{k+1} = y^k + \alpha^k \Delta y^k;$$

$$s^{k+1} = s^k + \alpha^k \Delta s^k;$$

$k=k+1$;

Fin de tant que

Fin d'algorithme.

Résultats de convergence

Théorème 2.8. (Théorème de convergence)

Soient σ et δ deux constantes vérifiant les conditions (2.7) et (2.8), $\hat{\mu} > 0$ défini par $\hat{\mu} = \mu(1 - \frac{\delta}{\sqrt{n}})$ où $\mu = \frac{(x^T s)}{n}$. Supposons que $(x, y, s) \in S_{cent(\sigma)}$ et soit $w_+ = (x_+, y_+, s_+)$ donnée par $w_+ = w + \alpha \Delta w$, alors nous avons

1. $\|X_+ s_+ - \hat{\mu} e\| \leq \sigma \hat{\mu}$.
2. $w_+ \in W$.
3. $g(w_+) = x_+^T s_+ = n \hat{\mu}$.

Corollaire 2.9. La suite de point w^k engendrée par l'algorithme satisfait

1. $\|X^k S^k e - \mu_k e\| \leq \sigma \mu_k$.
2. $w^k \in W, k = 1, \dots, n$.
3. $g(w^k) = (x^k)^T s^k = n \mu_k, k = 1, \dots, n$.

Où $\mu_k = \mu_0(1 - \frac{\delta}{\sqrt{n}})^k$.

Proposition 2.10. (Complexité arithmétique)

Le nombre total d'itérations effectuées par l'algorithme ne dépasse pas la valeur

$$k = \frac{\sqrt{n}}{\delta} \ln(n \epsilon^{-1} \mu_0).$$

Corollaire 2.11. Si le paramètre de pénalité initial μ_0 satisfait $\mu_0 = 2^{O(L)}$ alors l'algorithme résout les problèmes (P) et (D) dans $O(\sqrt{n}L)$ itérations et $O(n^{3.5}L)$ opérations arithmétiques.

L est une constante liée à la taille du problème (P) définie par

$$L = \left[\left(\begin{array}{c} \text{la plus grande valeur absolue du déterminant} \\ \text{d'une sous matrice de } A \end{array} \right) + 1 \right] \\ + [\log(1 + \max_i |c_i|)] + [\log(1 + \max_i |b_i|)] + [\log(m + n)].$$

Chapitre 3

Méthode de trajectoire centrale via les fonctions noyaux

Dans ce chapitre, on va étudier la méthode de trajectoire centrale primale-duale basée sur la notion des fonctions noyaux. A cet effet, on a choisi de détailler l'article de M. Bouafia, D. Benterki et A. Yassine, intitulé : "*An Efficient Primal-Dual Interior Point Method For Linear Programming Problems Based On A New Kernel Function With A Trigonometric Barrier Term*", dont les auteurs ont introduit une fonction noyau à terme barrière de type trigonométrique, pour trouver une nouvelle classe de directions dans le but d'obtenir un nouveau résultat de complexité.

Rappelons que le problème de la programmation linéaire, considéré comme primal est défini par

$$(P) \quad \begin{cases} \min \langle c, x \rangle \\ Ax = b, \\ x \geq 0, \end{cases}$$

où $A \in \mathbb{R}^{m \times n}$, $rg(A) = m$, $b \in \mathbb{R}^m$ et $\langle c, x \rangle = c^T x = \sum_{i=1}^n c_i x_i$, $c \in \mathbb{R}^n$.

Le problème dual associé à (P) est défini par

$$(D) \quad \begin{cases} \max \langle b, y \rangle \\ A^T y + s = c, \\ s \geq 0, y \in \mathbb{R}^m. \end{cases}$$

3.1 Méthode (TC) primale-duale basée sur les fonctions noyaux

Rappelons que les méthodes de trajectoires centrales primales-duales reposent sur les principes suivants

- On associe au problème primal (P) le problème perturbé (P_μ).
- On applique les conditions **KKT** sur ce dernier, on obtient le système non linéaire suivant

$$\begin{cases} A^T y + s - c = 0, \\ Ax - b = 0, \\ XSe - \mu e = 0, \\ s > 0, x > 0, \end{cases} \quad (3.1)$$

où on suppose que les ensembles S_{int} et T_{int} sont non vide cette condition est appelée aussi condition de point intérieur (CPI), sous cette condition (CPI) le système (3.1) admet une solution unique $(x(\mu), y(\mu), s(\mu))$ pour chaque $\mu > 0$.

- On résout le système (3.1) par la méthode Newton, cette dernière consiste à trouver la direction $\Delta w = (\Delta x, \Delta y, \Delta s)$, en résolvant le système linéaire suivant

$$\begin{cases} A\Delta x = 0, \\ A^T \Delta y + \Delta s = 0, \\ s\Delta x + x\Delta s = \mu e - xs. \end{cases} \quad (3.2)$$

L'itéré de Newton s'écrit comme suit

$$x_+ = x + \alpha\Delta x, \quad s_+ = s + \alpha\Delta s, \quad y_+ = y + \alpha\Delta y, \quad (3.3)$$

où α est un paramètre positif ($\alpha > 0$) introduit d'une façon à maintenir la stricte faisabilité des nouveaux itérés x_+ et s_+ .

3.1.1 Introduction de la notion des fonctions noyaux.

Pour faciliter l'étude de ces méthodes, on aura besoin de définir le vecteur v comme suit

$$v = \sqrt{\frac{xs}{\mu}}, \quad xs = (x_1 s_1, x_2 s_2, \dots, x_n s_n), \quad (3.4)$$

où v est appelé "le vecteur mis à l'échelle" (en anglais scaling vector).

En utilisant (3.4), le système (3.1) peut réécrire comme suit

$$\begin{cases} \bar{A}d_x = 0, \\ \bar{A}^T \Delta y + d_s = 0 \\ d_x + d_s = v^{-1} - v. \end{cases} \quad (3.5)$$

Où

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

et

$$d_x = \frac{v\Delta x}{x}, \quad d_s = \frac{v\Delta s}{s}, \quad (3.6)$$

d_x et d_s sont appelées "les directions de Newton mise à l'échelle" (en anglais scaling directions).

Notons que le côté droit de la troisième équation de (3.5) est égal au gradient négatif de la fonction barrière logarithmique $\Phi(v)$, i.e.,

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

dans ce cas, (3.5) devient

$$\begin{cases} \bar{A}d_x = 0, \\ \bar{A}^T \Delta y + d_s = 0, \\ d_x + d_s = -\nabla\Phi(v). \end{cases} \quad (3.7)$$

Où la fonction barrière $\Phi(v)$ est définie de \mathbb{R}_{++}^n dans \mathbb{R}_+^n par

$$\Phi(v) = \Phi(x, s; \mu) = \sum_{i=1}^n \Psi_c(v_i), \quad (3.8)$$

telle que Ψ_c est appelée la fonction noyau classique de la fonction barrière logarithmique $\Phi(v)$, définie par

$$\Psi_c(t) = \frac{t^2 - 1}{2} - \log t. \quad (3.9)$$

Remarque. Si $(d_x, \Delta y, d_s)$ est une solution du système (3.7), alors d_x et d_y sont orthogonaux.

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

$$\delta(v) = \frac{1}{2} \|\nabla\Phi(v)\| = \frac{1}{2} \|d_x + d_s\|. \quad (3.10)$$

3.1.2 Algorithme prototype de la méthode (TC) primale-duale

Début algorithme

Données

Φ La fonction de proximité ;

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

$\epsilon > 0$, un paramètre de précision ;

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

Initialisation :

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

Tant que $n\mu \geq \epsilon$ **faire**

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

Tant que $\Phi(x, y, s) > \tau$ **faire**

— Résoudre le système (3.7), et utiliser (3.6) pour déterminer $(\Delta x, \Delta y, \Delta s);$

— Calculer le pas de déplacement $\alpha;$

— Poser : $x := x + \alpha\Delta x; y := y + \alpha\Delta y; s := s + \alpha\Delta s;$

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

Fin tant que

Fin tant que

Fin algorithme.

3.2 Nouvelle fonction noyau

Dans ce qui suit, au lieu d'utiliser la fonction noyau classique Ψ_c , on utilise la fonction noyau proposée par Bouafia et autres [2] comme suit

$$\Psi(t) = \frac{t^2 - 1}{2} + \frac{4}{\pi p} [\tan^p h(t) - 1], h(t) = \frac{\pi}{2t + 2}, p \geq 2. \quad (3.11)$$

3.2.1 Propriétés de la nouvelle fonction noyau

Dans cette section, nous étudions certaines propriétés de la nouvelle fonction noyau à terme barrière trigonométrique, qui sont essentielles pour l'analyse de la complexité algorithmique. Avant d'étudier les propriétés de cette nouvelle fonction noyau, nous avons besoin de donner la définition d'une fonction noyau.

Définition 3.1. Soit $\Psi(t) : \mathbb{R}_{++} \longrightarrow \mathbb{R}_+$, une fonction deux fois continûment différentiable. Alors Ψ est dite fonction noyau, si elle vérifie les conditions suivantes

1. $\Psi'(1) = \Psi(1) = 0$,
2. $\Psi''(t) > 0$,
3. $\lim_{t \rightarrow 0^+} \Psi(t) = \lim_{t \rightarrow +\infty} \Psi(t) = +\infty$.

Les deux premières conditions montrent que Ψ est strictement convexe et minimale en 1 avec $\Psi(1) = 0$ et implique que $\Psi(t)$ s'écrit comme suit

$$\Psi(t) = \int_1^t \left(\int_1^x \Psi''(y) dy \right) dx. \quad (3.12)$$

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

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

$$\Psi'(t) = t + \frac{4}{\pi} \sec^2 h(t) (\tan^{p-1} h(t)) h'(t), \quad (3.13)$$

$$\Psi''(t) = 1 + \frac{4}{\pi} \sec^2 h(t) \begin{bmatrix} \left[\begin{array}{l} (p-1) \tan^{p-2} h(t) + \\ (p+1) \tan^p h(t) \end{array} \right] (h'(t))^2 + \\ [\tan^{p-1} h(t)] h''(t) \end{bmatrix}, \quad (3.14)$$

$$\Psi'''(t) = \frac{4}{\pi} \sec^2 h(t) \begin{bmatrix} \left[\begin{array}{l} (p-1)(p-2) \tan^{p-3} h(t) + \\ 2p^2 \tan^{p-1} h(t) + \\ (p+1)(p+2) \tan^{p+1} h(t) \end{array} \right] (h'(t))^3 + \\ \left[\begin{array}{l} 3(p-1) \tan^{p-2} h(t) + \\ 3(p+1) \tan^p h(t) \end{array} \right] h''(t) h'(t) + \\ [\tan^{p-1} h(t)] h'''(t) \end{bmatrix}, \quad (3.15)$$

avec

$$h'(t) = \frac{-\pi}{2(t+1)^2}; h''(t) = \frac{\pi}{(t+1)^3}; h'''(t) = \frac{-3\pi}{(t+1)^4}, \quad (3.16)$$

et

$$\sec h(t) = \frac{1}{\cos h(t)}.$$

3.2.2 Quelques fonctions noyaux et leur complexité

Dans le tableau suivant, on donne les différentes fonctions noyaux connues dans la littérature [1, 2] et la complexité de leur algorithme pour les méthodes de point intérieur à petit- et à grand-pas.

	Fonction noyau	Complexité algorithmique	Complexité algorithmique
	$\Psi_i(t)$	Petit-pas	Grand-pas
1	$\frac{1}{2}(t^2 - 1) - \log t$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n \log \frac{n}{\varepsilon})$
2	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q(q-1)} - \frac{q-1}{q(q-1)}(t-1), q > 1$	$O(q\sqrt{n} \log \frac{n}{\varepsilon})$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$
3	$\frac{t^2-1}{2} + \frac{(e-1)^2}{e} \frac{1}{e-1} - \frac{e-1}{e}$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$
4	$\frac{1}{2}(t - \frac{1}{t})^2$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n^{\frac{2}{4}} \log \frac{n}{\varepsilon})$
5	$\frac{t^2-1}{2} + e^{\frac{1}{t-1}} - 1$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(\sqrt{n}(\log n)^2 \log \frac{n}{\varepsilon})$
6	$\frac{t^2-1}{2} - \int_1^t e^{\frac{1}{\xi}} d\xi$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon})$
7	$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{(q-1)}, q > 1$	$O(q^2 \sqrt{n} \log \frac{n}{\varepsilon})$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$
8	$t - 1 + \frac{t^{1-q}-1}{q-1}, q > 1$	$O(q^2 \sqrt{n} \log \frac{n}{\varepsilon})$	$O(qn \log^2 n \log \frac{n}{\varepsilon})$
9	$\frac{t^{1+p}-1}{p+1} - \log t, p \in [0, 1]$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n \log \frac{n}{\varepsilon})$
10	$\frac{t^{1+p}-1}{p+1} + \frac{t^{1-q}-1}{q-1}, p \in [0, 1], q > 1$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(qn^{\frac{p+q}{q(1+p)}} \log \frac{n}{\varepsilon})$
11	$t^2 - 1 + \frac{t^{1-q}-1}{q-1} - \log t, p > 1, q > 1$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$
12	$(m+1)t^2 - (m+2)t + \frac{1}{t^m}, m > 4$	$O(q^2 \sqrt{n} \log \frac{n}{\varepsilon})$	$O(m^{\frac{2m+1}{2m}} \log \frac{n}{\varepsilon})$
13	$t^2 - 2t + \frac{1}{\sin(\frac{\pi t}{1+t})}, t > 0$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$
14	$\frac{t^2-1}{2} + \frac{4}{\pi p} [\tan^p h(t) - 1], t > 0$	$O(p^2 \sqrt{n} \log \frac{n}{\varepsilon})$	$O(pn^{\frac{p+2}{2(p+1)}} \log \frac{n}{\varepsilon})$

TABLE 3.1 – La complexité à petit- et à grand-pas

Lemme 3.2. Soit $\Psi(t)$ une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes

1. $\Psi(\sqrt{t_1 t_2}) \leq \frac{\Psi(t_1) + \Psi(t_2)}{2}$, pour tout $t_1, t_2 > 0$.
2. $t\Psi''(t) + \Psi'(t) \geq 0$, $t > 0$.
3. $\Psi(e^\varepsilon)$ est convexe.

Lemme 3.3. Pour $h(t)$ définie dans (3.11) et $p \geq 2$, nous avons les propriétés suivantes

$$0 < h(t) < \frac{\pi}{2}, t > 0. \quad (3.17)$$

$$\tan h(t) > 0, t > 0. \quad (3.18)$$

$$\tan h(t) - \frac{2}{(p+1)\pi t} > 0, t > 0. \quad (3.19)$$

Démonstration. Pour (3.17), tant que $h(t)$ est une fonction décroissante dans l'intervalle $]0, +\infty[$ et $\lim_{t \rightarrow 0^+} h(t) = \frac{\pi}{2}$, $\lim_{t \rightarrow +\infty} h(t) = 0$. Alors, on a

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

(3.18) est une conséquence immédiate de (3.17).

Pour prouver (3.19), on définit la fonction g comme suit

$$g(t) = \tan h(t) - \frac{2}{(p+1)\pi t},$$

la première dérivée de g est donnée par

$$g'(t) = \frac{h'(t)}{\cos^2 h(t)} + \frac{2}{(p+1)\pi t} = \frac{(p+1)\pi t^2 h'(t) + 2 \cos^2 h(t)}{(p+1)\pi t^2 \cos^2 h(t)}.$$

On sait que $\sin(\frac{\pi}{2} - h(t)) = \cos h(t)$ et $\sin(\frac{\pi}{2} - h(t)) \leq (\frac{\pi}{2} - h(t))$ et par l'utilisation de (3.17), il s'en suit que

$$\begin{aligned} g'(t) &= \frac{(p+1)\pi t^2 h'(t) + 2 \cos^2(\frac{\pi}{2})}{(p+1)\pi t^2 \cos^2 h(t)} \\ &= \frac{(p+1)\pi t^2 h'(t) + 2 \sin^2(\frac{\pi}{2} - h(t))}{(p+1)\pi t^2 \cos^2 h(t)} \\ &\leq \frac{(p+1)\pi t^2 h'(t) + 2(\frac{\pi}{2} - h(t))^2}{(p+1)\pi t^2 \cos^2 h(t)} \\ &= \frac{\frac{-(p+1)\pi^2 t^2}{2(t+1)^2} + 2\left(\frac{\pi^2}{4} - 2\frac{\pi}{2}\frac{\pi}{2t+2} + \frac{\pi^2}{4(t+1)^2}\right)}{(p+1)\pi t^2 \cos^2 h(t)} \\ &= \frac{-2p\pi}{(p+1)\cos^2 h(t)(2t+2)^2} < 0 \end{aligned}$$

Donc, g est une fonction décroissante dans l'intervalle $]0, +\infty[$ et comme $\lim_{t \rightarrow +\infty} g(t) = 0$, on arrive à (3.19).

Ce qui complète la preuve. ■

Le lemme suivant sert à prouver l'efficacité de notre nouvelle fonction noyau (3.11).

Lemme 3.4. *Soit $\Psi(t)$ la fonction noyau définie dans (3.11). Alors, pour tout $t > 0$, on a*

$$\Psi''(t) > 1. \tag{3.20}$$

$$\Psi'''(t) < 0. \tag{3.21}$$

$$t\Psi''(t) - \Psi'(t) > 0. \tag{3.22}$$

$$t\Psi''(t) + \Psi'(t) > 0. \tag{3.23}$$

Démonstration. Pour (3.20), en utilisant (3.16) et (3.18) ce qui implique que tous les termes de l'expression (3.14) sont positifs, ce qui donne (3.20).

Pour prouver (3.21), en utilisant (3.15), (3.16) et (3.18) on obtient, pour tout $t >$

$0, \Psi'''(t) < 0$.

Pour (3.22), nous utilisons (3.13) et (3.14), nous avons pour tout $t > 0$,

$$t\Psi''(t) - \Psi'(t) = \frac{4}{\pi} \sec^2 h(t) \left[\begin{array}{l} (h'(t))^2 [(p-1) \tan^{p-2} h(t) + (p+1) \tan^p h(t)] t + \\ h''(t) (\tan^{p-1} h(t)) t + (\tan^{p-1} h(t)) (-h'(t)) \end{array} \right]$$

de (3.16) tous les termes de la dernière égalité sont positifs, ce qui prouve (3.22).

Pour la dernière inégalité (3.23), en utilisant à nouveau (3.13) et (3.14), on trouve,

$$t\Psi''(t) + \Psi'(t) = 2t + \frac{4}{\pi} \sec^2 h(t) \left[\begin{array}{l} (h'(t))^2 [(p-1) \tan^{p-2} h(t)] t + \\ (\tan^{p-1} h(t)) \left[\begin{array}{l} ((p+1)(h'(t))^2 \tan h(t)) t + \\ h''(t)t + h'(t) \end{array} \right] \end{array} \right]$$

avec

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

d'après (3.19), on a le côté droit de l'égalité au dessus est positif ce qui prouve (3.23).

Ce qui complète la preuve de ce Lemme. ■

Lemme 3.5. Soit $\Psi(t)$ la fonction noyau définie dans (3.11), si $\Psi(t)$ vérifie les propriétés (3.20) et (3.22), alors

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

Lemme 3.6. Pour $\Psi(t)$, la fonction noyau définie dans (3.11), nous avons

$$\frac{1}{2}(t-1)^2 \leq \Psi(t) \leq \frac{1}{2}[\Psi'(t)]^2, \quad t > 0. \quad (3.24)$$

$$\Psi(t) \leq \left(1 + \frac{\pi}{8}p\right) (t-1)^2, \quad t > 1. \quad (3.25)$$

Démonstration. Pour (3.24), en utilisant (3.20) dans (3.12), on obtient

$$\begin{aligned} \Psi(t) &= \int_1^t \int_1^x \Psi''(y) dy dx \geq \int_1^t \int_1^x dy dx \\ &= \int_1^t (x-1) dx \\ &= \frac{1}{2} [(x-1)^2]_1^t \\ &= \frac{1}{2} (t-1)^2, \end{aligned}$$

d'où le côté gauche de l'inégalité (3.24). De même pour le côté droit de cette inégalité, on trouve

$$\begin{aligned}
\Psi(t) &= \int_1^t \int_1^x \Psi''(y) dy dx \leq \int_1^t \int_1^x \Psi''(y) \Psi''(x) dy dx \\
&= \int_1^t \Psi''(x) [\Psi'(y)]_1^x dx \\
&= \int_1^t \Psi''(x) \Psi'(x) dx \\
&= \frac{1}{2} [\Psi'(t)]^2.
\end{aligned}$$

Pour (3.25), on utilise le développement de Taylor en $t = 1$ avec $\Psi(1) = \Psi'(1) = 0$, $\Psi''(1) = (2 + \frac{\pi}{4}p)$ et $\Psi'''(t) < 0$, on trouve

$$\begin{aligned}
\Psi(t) &= \Psi(1) + \Psi'(1)(t-1) + \frac{\Psi''(1)}{2}(t-1)^2 + \frac{\Psi'''(\xi)}{6}(t-1)^3, \quad 1 \leq \xi \leq t \\
&= \frac{\Psi''(1)}{2}(t-1)^2 + \frac{\Psi'''(t)}{6}(\xi-1)^3, \quad 1 \leq \xi \leq t \\
&\leq \frac{(2 + \frac{\pi}{4}p)(t-1)^2}{2} \\
&= \left(1 + \frac{\pi}{8}p\right)(t-1)^2.
\end{aligned}$$

■

Lemme 3.7. Soient $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de la fonction $\Psi(t)$ pour $t \geq 1$ et $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de la fonction $-\frac{1}{2}\Psi'(t)$ pour $t \in]0, 1]$, alors on a

$$1 + \sqrt{\frac{8}{8 + \pi p}} s \leq \sigma(s) \leq 1 + \sqrt{2s}, \quad s \geq 0, \quad (3.26)$$

$$\tan h(t) \leq (4z + 2)^{\frac{1}{p+1}}, \quad z \geq 0. \quad (3.27)$$

Démonstration. Pour (3.26), soit $s = \Psi(t)$, $t \geq 1$, i.e., $\sigma(s) = t$, $t \geq 1$.

D'après (3.24), on a $\Psi(t) \geq \frac{1}{2}(t-1)^2$,

ce qui implique

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

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

$$t-1 = |t-1| \leq \sqrt{2s},$$

d'où, d'une part

$$\sigma(s) = t \leq 1 + \sqrt{2s}, \quad t \geq 1.$$

D'autre part et d'après (3.25), on a

$$s = \Psi(t) \leq \left(1 + \frac{\pi}{8}p\right) (t-1)^2, \quad t > 1,$$

ce qui donne

$$\frac{8s}{(8 + \pi p)} \leq (t-1)^2,$$

ce qui implique

$$t-1 = |t-1| \geq \sqrt{\frac{8s}{8 + \pi p}},$$

d'où

$$t = \sigma(s) \geq 1 + \sqrt{\frac{8s}{8 + \pi p}}.$$

Pour (3.27), soit $z = -\frac{1}{2}\Psi'(t)$, $t \in]0, 1]$ qui est équivalent à

$$2z = -\Psi'(t), t \in]0, 1].$$

De (3.13), on trouve

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

Ce qui implique

$$(1 + \tan^2 h(t)) \tan^{p-1} h(t) = \frac{-\pi(2z + t)}{4h'(t)} \leq 2(2z + 1), \quad \forall t \in]0, 1],$$

et comme

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

on trouve

$$\tan^{p+1} h(t) \leq 2(2z + 1).$$

D'où

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

Ce qui achève la preuve. ■

Lemme 3.8. *Supposons que $\Psi(t_1) = \Psi(t_2)$ avec $t_1 \leq 1 \leq t_2$ et $\beta \geq 1$, alors*

$$\Psi(\beta t_1) \leq \Psi(\beta t_2).$$

L'égalité est vérifiée si et seulement si $\beta = 1$ ou $t_1 = t_2 = 1$.

Le Lemme suivant est un Lemme très important, qui est valide pour toute fonction noyau vérifie les conditions (3.20) et (3.21).

Lemme 3.9. [1] Soit $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de la fonction $\Psi(t)$, $t \geq 1$, alors on a

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

Démonstration. Pour $\beta > 1$, nous considérons le problème de maximisation suivant

$$\text{Max } \{\Phi(\beta v) : \Phi(v) = z, \forall z > 0\}.$$

Les conditions d'optimalité du premier ordre (ou le lagrangien) donne

$$\Phi(\beta v) + y(z - \Phi(v)) = 0, \quad (1)$$

où $y > 0$ est le multiplicateur de Lagrange. En dérivant (1) par rapport à v , on obtient

$$\beta\Phi'(\beta v) - y\Phi'(v) = 0,$$

ou de manière équivalente

$$\beta \sum_{i=1}^n \Psi'(\beta v_i) - y \sum_{i=1}^n \Psi'(v_i) = 0 \iff \sum_{i=1}^n \beta \Psi'(\beta v_i) - \sum_{i=1}^n y \Psi'(v_i) = 0,$$

ce qui est équivalent à

$$\beta \Psi'(\beta v_i) = y \Psi'(v_i), \forall i = 1, \dots, n. \quad (2)$$

Comme $\Psi'(1) = 0$ et $\Psi'(\beta) > 0, \forall \beta > 1$, alors $\Psi'(v_i) > 0$

Donc $v_i \neq 1$, pour v_i vérifie les contraintes du problème (2), i.e., $\Phi(v_i) = z_i, \forall i = \overline{1, n}$, avec z_i donné, on distingue deux types de valeurs de v_i

$$v_i = \begin{cases} v_i^+, & \text{si } v_i > 1, \\ v_i^-, & \text{si } v_i < 1, \end{cases}$$

donc

$$v_i^- < 1 < v_i^+.$$

D'après le Lemme 3.8, on a

$$\Psi(\beta v_i^-) \leq \Psi(\beta v_i^+),$$

donc, on prend

$$v_i = v_i^+ > 1, \forall i = 1, \dots, n.$$

Puisque nous avons un problème de maximisation, l'équation (2) donne

$$\beta \Psi'(\beta v_i) > 0 \text{ et } y \Psi'(v_i) > 0, \quad y > 0.$$

Maintenant, on définit la fonction g comme suit

$$g(t) = \frac{\Psi'(t)}{\Psi'(\beta t)}, \quad t \geq 1,$$

d'après l'équation (2), on déduit que

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

Donc g est une fonction constante.

On dérive g on obtient

$$g'(t) = \frac{\Psi''(t)\Psi'(\beta t) - \beta\Psi'(t)\Psi''(\beta t)}{(\Psi'(\beta t))^2}, \quad \forall \beta > 1, \quad \forall t \geq 1.$$

D'après le Lemme 3.5, on trouve que $g'(t) > 0$. Donc, g est une fonction strictement croissante, d'où tous les v_i sont égaux.

Posons $v_i = t$, $\forall i = 1, \dots, n$, de l'égalité $\Phi(v) = z$, on déduit que

$$\sum_{i=1}^n \Psi(t) = n\Psi(t) = z \iff \Psi(t) = \frac{z}{n}.$$

Comme σ est la fonction inverse de la fonction Ψ , alors $t = \sigma\left(\frac{z}{n}\right)$. D'où

$$\Phi(\beta v) \leq \text{Max } \Phi(\beta v) = \Phi(\beta t) = n\Psi(\beta t) = n\Psi\left(\beta\sigma\left(\frac{z}{n}\right)\right) = n\Psi\left(\beta\sigma\left(\frac{\Phi(v)}{n}\right)\right).$$

Donc

$$\Phi(\beta v) \leq n\Psi\left(\beta\sigma\left(\frac{\Phi(v)}{n}\right)\right), \quad \forall \beta > 1.$$

Le cas $\beta = 1$, est évident puisque

$$\frac{\Phi(v)}{n} = \Psi\left(\sigma\left(\frac{\Phi(v)}{n}\right)\right).$$

Ce qui implique

$$\Phi(v) = n\Psi\left(\sigma\left(\frac{\Phi(v)}{n}\right)\right).$$

■

Le théorème suivant, nous a permis d'obtenir l'estimation de Φ après une μ mise à jour.

Théorème 3.10. Soit $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$, si $\Phi(v) \leq \tau$, alors on a

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

Démonstration. On a $\sigma : [0, +\infty[\rightarrow [1, +\infty[$ la fonction inverse de $\Psi(t)$, $t \geq 1$, donc $\sigma\left(\frac{\Phi(v)}{n}\right) \geq 1$ et comme $0 \leq \theta \leq 1$, on trouve $\frac{1}{\sqrt{1-\theta}} \geq 1$.

Ce qui implique

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

pour $t \geq 1$, on a

$$\Psi(t) \leq \frac{t^2 - 1}{2}. \quad (*)$$

Par l'utilisation du Lemme 3.9 avec $\beta = \frac{1}{\sqrt{1-\theta}}$, on obtient

$$\Phi(v_+) \leq n\Psi\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi(v)}{n}\right)\right),$$

ce qui donne, d'après (*)

$$\begin{aligned} \Phi(v_+) &\leq \frac{n}{2} \left(\left(\frac{1}{\sqrt{1-\theta}} \sigma\left(\frac{\Phi(v)}{n}\right) \right)^2 - 1 \right) \\ &= \frac{n}{2} \left(\frac{1}{1-\theta} \left(\sigma\left(\frac{\Phi(v)}{n}\right) \right)^2 - 1 \right) \\ &= \frac{n}{2} \left(\frac{1}{(1-\theta)} \left(\sigma\left(\frac{\Phi(v)}{n}\right) \right)^2 - 1 \right) \\ &= \frac{n}{2(1-\theta)} \left(\sigma\left(\frac{\Phi(v)}{n}\right)^2 - (1-\theta) \right), \end{aligned}$$

de (3.26), on a

$$\begin{aligned} \Phi(v_+) &\leq \frac{n}{2(1-\theta)} \left(\left(1 + \sqrt{2\frac{\Phi(v)}{n}} \right)^2 - (1-\theta) \right) \\ &= \frac{n}{2(1-\theta)} \left(\theta + 2\frac{\Phi(v)}{n} + 2\sqrt{2\frac{\Phi(v)}{n}} \right), \end{aligned}$$

comme $\Phi(v) \leq \tau$, on obtient

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

Ce qui termine la démonstration. ■

Notons par

$$(\Phi)_0 := \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)}. \quad (3.28)$$

Donc, $(\Phi)_0$ devient une borne supérieure de $\Phi(v_+)$ durant le processus de l'algorithme.

3.2.3 Analyse de la complexité

Calcul du pas de déplacement

Dans cette partie, nous calculons le pas de déplacement par défaut α et nous prouvons le décroissement de la fonction barrière. Après une itération, nous avons

$$x_+ = x + \alpha\Delta x ; y_+ = y + \alpha\Delta y ; s_+ = s + \alpha\Delta s.$$

Utilisons (3.4), nous avons

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

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

d'où

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

Pour tout $\alpha > 0$, on définit la fonction f comme suit

$$f(\alpha) = \Phi(v_+) - \Phi(v). \quad (3.29)$$

Donc, $f(\alpha)$ est la différence de proximité entre le nouvel et le courant itéré, pour μ fixe.

Par (1) du Lemme 3.2, nous avons

$$\Phi(v_+) = \Phi \left(\sqrt{(v + \alpha d_x)(v + \alpha d_s)} \right) \leq \frac{1}{2} (\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)),$$

et par conséquent $f(\alpha) \leq f_1(\alpha)$, telle que

$$f_1(\alpha) = \frac{1}{2} (\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)) - \Phi(v), \quad (3.30)$$

pour $\alpha = 0$, on a

$$f(0) = \Phi(\sqrt{v^2}) - \Phi(v) = \Phi(v) - \Phi(v) = 0,$$

et

$$f_1(0) = \frac{1}{2} (\Phi(v) + \Phi(v)) - \Phi(v) = 0,$$

d'où

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

De (3.8), on a

$$f_1(\alpha) = \frac{1}{2} \sum_{i=1}^n (\Psi(v_i + \alpha d_{x_i}) + \Psi(v_i + \alpha d_{s_i})) - \sum_{i=1}^n \Psi(v_i).$$

Prenant les deux premières dérivées de $f_1(\alpha)$ par rapport à α , nous obtenons

$$f_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n (\Psi'(v_i + \alpha d_{x_i}) d_{x_i} + \Psi'(v_i + \alpha d_{s_i}) d_{s_i}),$$

et

$$f_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n (\Psi''(v_i + \alpha d_{x_i}) d_{x_i}^2 + \Psi''(v_i + \alpha d_{s_i}) d_{s_i}^2),$$

où d_{x_i} et d_{s_i} désigne la $i^{\text{ème}}$ composante de vecteurs d_x et d_s , respectivement.

Pour $\alpha = 0$, on trouve

$$\begin{aligned} f_1'(0) &= \frac{1}{2} \sum_{i=1}^n \Psi'(v_i) d_{x_i} + \Psi'(v_i) d_{s_i} \\ &= \frac{1}{2} \sum_{i=1}^n \Psi'(v_i) (d_{x_i} + d_{s_i}). \end{aligned}$$

En la dernière équation de (3.7) et (3.8), on obtient

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

de (3.10), on a

$$f_1'(0) = -2\delta(v)^2.$$

Lemme 3.11. Soit $\delta(v)$ définie par (3.10), alors

$$\delta(v) \geq \sqrt{\frac{1}{2} \Phi(v)}. \quad (3.31)$$

Démonstration. En utilisant (3.24), nous avons

$$\Phi(v) = \sum_{i=1}^n \Psi(v_i) \leq \sum_{i=1}^n \frac{1}{2} (\Psi'(v_i))^2 = \frac{1}{2} \|\nabla \Phi(v)\|^2 = 2\delta(v)^2,$$

ce qui donne

$$\delta(v) \geq \sqrt{\frac{1}{2} \Phi(v)}.$$

Pour $\Phi(v) \geq 1$, on trouve $\delta(v) \geq \sqrt{\frac{1}{2}}$. ■

Remarque. *Tout au long de ce chapitre, nous supposons que $\tau > 1$. En utilisant le Lemme 3.11 et la supposition de $\Phi(v) \geq \tau$, nous avons à partir des Lemmes 4.1 – 4.4 de [1] les Lemmes 3.12 – 3.15 suivants*

Lemme 3.12. *Soient $f_1(\alpha)$ et $\delta(v)$ définie dans (3.30) et (3.10), respectivement. Alors*

$$f_1''(\alpha) \leq 2\delta^2\Psi''(v_1 - 2\alpha\delta),$$

où $\delta(v) := \delta$.

Démonstration. De la dernière égalité de (3.7) et par définition de δ , on a $\|d_x + d_s\| = 2\delta$, d'où

$$\|d_x + d_s\|^2 = \|d_x\|^2 + \|d_s\|^2 + 2\|d_x d_s\| = (2\delta)^2.$$

Puisque d_x et d_s sont orthogonaux, nous avons $\|d_x\| \leq 2\delta$ et $\|d_s\| \leq 2\delta$, i.e.,

$$\|d_x\| = \left(\sum_{i=1}^n d_{x_i}^2 \right)^{\frac{1}{2}} \leq 2\delta.$$

$$\|d_s\| = \left(\sum_{i=1}^n d_{s_i}^2 \right)^{\frac{1}{2}} \leq 2\delta.$$

Donc

$$-2\delta \leq |d_{x_i}| \leq 2\delta \text{ et } -2\delta \leq |d_{s_i}| \leq 2\delta, \quad 1 \leq i \leq n.$$

Alors

$$-2\alpha\delta \leq \alpha|d_{x_i}| \text{ et } -2\alpha\delta \leq \alpha|d_{s_i}|, \quad 1 \leq i \leq n.$$

On obtient

$$v_i + \alpha d_{x_i} \geq v_1 + \alpha d_{x_i} \geq v_1 - 2\alpha\delta$$

et

$$v_i + \alpha d_{x_i} \geq v_1 - \alpha d_{s_i} \geq v_1 - 2\alpha\delta, \quad 1 \leq i \leq n.$$

De (3.21), Ψ'' est strictement décroissante, donc la substitution de ces inégalités à f_1'' , nous donne

$$\begin{aligned} f_1''(\alpha) &= \frac{1}{2} \sum_{i=1}^n (\Psi''(v_i + \alpha d_{x_i}) d_{x_i}^2 + \Psi''(v_i + \alpha d_{s_i}) d_{s_i}^2) \\ &\leq \frac{1}{2} \sum_{i=1}^n (\Psi''(v_1 - 2\alpha\delta) d_{x_i}^2 + \Psi''(v_1 - 2\alpha\delta) d_{s_i}^2) \\ &= \frac{1}{2} \Psi''(v_1 - 2\alpha\delta) \sum_{i=1}^n (d_{x_i}^2 + d_{s_i}^2) \\ &= \frac{1}{2} \Psi''(v_1 - 2\alpha\delta) \|d_x + d_s\|^2 = 2\delta^2 \Psi''(v_1 - 2\alpha\delta). \end{aligned}$$

Ce qui complète la preuve. ■

Lemme 3.13. *Si le pas de déplacement α vérifie l'inégalité suivante*

$$\Psi'(v_1) - \Psi'(v_1 - 2\alpha\delta) \leq 2\delta, \quad (3.32)$$

alors

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

Démonstration. On a

$$f'_1(\alpha) = f'_1(0) + \int_0^\alpha f''_1(\xi) d\xi,$$

comme $f'_1(0) = -2\delta^2$ et d'après le Lemme 3.12, nous avons

$$\begin{aligned} f'_1(\alpha) &= f'_1(0) + \int_0^\alpha f''_1(\xi) d\xi \\ &\leq -2\delta^2 + 2\delta^2 \int_0^\alpha \Psi''(v_1 - 2\xi\delta) d\xi \\ &= -2\delta^2 + \delta(-\Psi'(v_1 - 2\alpha\delta) + \Psi'(v_1)) \\ &\leq 0. \end{aligned}$$

Ce qui termine la preuve. ■

Lemme 3.14. *Soit $\rho : [0, +\infty[\rightarrow]0, 1]$ la fonction inverse de la fonction $[-\frac{1}{2}\Psi'(t)]$, pour $t \in]0, 1]$, alors la valeur maximale du pas de déplacement α qui vérifie l'inégalité (3.32) est donnée par*

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

Lemme 3.15. *Soient ρ et $\bar{\alpha}$ le pas de déplacement défini dans le Lemme 3.14, alors*

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

Démonstration. Par définition de ρ , on a

$$\delta = -\frac{1}{2}\Psi'(\rho(\delta)),$$

qui est équivalent à

$$2\delta = -\Psi'(\rho(\delta)).$$

On dérive par rapport à δ , on trouve

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

ce qui implique

$$\rho'(\delta) = -\frac{2}{\Psi''(\rho(\delta))} < 0.$$

Donc, ρ est une fonction strictement décroissante dans $[0, +\infty[$.

D'après le Lemme 3.14, on a

$$\begin{aligned}\bar{\alpha} &= \frac{1}{2\delta} \int_{2\delta}^{\delta} \rho'(\xi) d\xi \\ &= -\frac{1}{\delta} \int_{2\delta}^{\delta} \frac{d\xi}{\Psi''(\rho(\delta))}.\end{aligned}$$

Comme ρ et Ψ'' sont des fonctions décroissantes, alors

$$\Psi''(\rho(\delta)) = \min_{\xi \in [\delta, 2\delta]} \Psi''(\rho(\xi)).$$

On sait que $\Psi''(\rho(\delta)) > 0$ et Ψ'' est décroissante, donc le maximum est atteint lorsque ρ est minimale, i.e.,

$$\Psi''(\rho(2\delta)) = \max_{\xi \in [\delta, 2\delta]} \Psi''(\rho(\xi)),$$

d'où

$$\bar{\alpha} \geq \frac{1}{\delta} \frac{1}{\Psi''(\rho(2\delta))} \int_{\delta}^{2\delta} d\xi = \frac{1}{\Psi''(\rho(2\delta))}.$$

■

Lemme 3.16. *Soient ρ et $\bar{\alpha}$ définies dans le Lemme 3.15. Si $\Phi(v) \geq \tau \geq 1$, alors nous avons*

$$\bar{\alpha} \geq \frac{1}{(9 + 4p\pi)(8\delta + 2)^{\frac{p+2}{p+1}}}.$$

Démonstration. En utilisant le Lemme 3.15, pour $\rho(2\delta) = t$, $t \in]0, 1]$, nous avons

$$h'(t)^2 \leq \left(\frac{\pi}{2}\right)^2; \quad h''(t) \leq \pi.$$

D'après (3.27), pour $z = 2\delta$, nous avons

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

et

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

Pour tout $t \in]0, 1]$, on a $\frac{\pi}{4} \leq h(t) < \frac{\pi}{2}$,

Ce qui implique

$$1 < \frac{1}{\sin^2 h(t)} \leq 2.$$

En remplaçant par ces inégalités, on trouve

$$\begin{aligned}\Psi''(t) &= 1 + \frac{4}{\pi} \sec^2 h(t) \left[\begin{array}{c} \left[\begin{array}{c} (p-1) \tan^{p-2} h(t) + \\ (p+1) \tan^p h(t) \end{array} \right] (h'(t))^2 + \\ [\tan^{p-1} h(t)] h''(t) \end{array} \right] \\ &\leq 1 + \tan^p \left[\frac{4}{\pi} \sec^2 h(t) \left[2p \left(\frac{-\pi}{2(t+1)^2} \right)^2 + \frac{\pi}{(t+1)^3} \right] \right] \\ &\leq \left[1 + \frac{4}{\pi} (2) \left(2p \left(\frac{\pi}{2} \right)^2 + \pi \right) \right] (8\delta + 2)^{\frac{p+2}{p+1}}.\end{aligned}$$

Ce qui donne

$$\bar{\alpha} \geq \frac{1}{\left[1 + \frac{4}{\pi} (2) \left(2p \left(\frac{\pi}{2} \right)^2 + \pi \right) \right] (8\delta + 2)^{\frac{p+2}{p+1}}} = \frac{1}{(9 + 4\pi p)(8\delta + 2)^{\frac{p+2}{p+1}}}.$$

■

Notons par

$$\tilde{\alpha} := \frac{1}{(9 + 4\pi p)(8\delta + 2)^{\frac{p+2}{p+1}}}. \quad (3.33)$$

Donc, $\tilde{\alpha}$ est la borne inférieure du pas de déplacement par défaut et on a

$$\tilde{\alpha} \leq \bar{\alpha}.$$

Lemme 3.17. *On suppose que g est une fonction convexe et deux fois différentiable avec*

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

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

$$g(t) \leq \frac{tg'(0)}{2}.$$

Lemme 3.18. [1] *Si la valeur du pas de déplacement α satisfait $\alpha \leq \bar{\alpha}$, alors*

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

Démonstration. Soit g une fonction définie par

$$g(\alpha) = -2\alpha\delta^2 + \alpha\delta\Psi'(v_1) - \frac{1}{2}\Psi(v_1) + \frac{1}{2}\Psi(v_1 - 2\alpha\delta),$$

alors

$$g(0) = 0 = f_1(0).$$

Par dérivation par rapport à α , on trouve

$$g'(\alpha) = -2\delta^2 + \delta\Psi'(v_1) - \delta\Psi'(v_1 - 2\alpha\delta).$$

D'après (3.32), on a

$$g'(\alpha) \leq -2\delta^2 + 2\delta^2 = 0.$$

Pour $\alpha = 0$, on trouve

$$g'(0) = -2\delta^2 = f'_1(0).$$

La dérivée seconde de $g(\alpha)$, est donnée par

$$g''(\alpha) = 2\delta^2\Psi''(v_1 - 2\alpha\delta).$$

D'après le Lemme 3.12

$$g''(\alpha) \geq f''_1(\alpha),$$

et par conséquent

$$\begin{cases} f'_1(\alpha) \leq g'(\alpha) \\ f_1(\alpha) \leq g(\alpha). \end{cases}$$

Alors, si $\alpha \leq \bar{\alpha}$, on a

$$\begin{aligned} g'(\alpha) &= g'(0) + \int_0^\alpha g''(\xi)d\xi \\ &= -2\delta^2 + 2\delta^2 \int_0^\alpha \Psi''(v_1 - 2\xi\delta)d\xi \\ &= -2\delta^2 - \delta \int_0^\alpha \Psi''(v_1 - 2\xi\delta)d(v_1 - 2\xi\delta) \\ &= -2\delta^2 + \delta(-\Psi'(v_1 - 2\alpha\delta) + \Psi'(v_1)). \end{aligned}$$

De (3.29) et (3.30), on a g est une fonction convexe et deux fois différentiable alors d'après le Lemme 3.17, on a

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

D'où

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

Ce qui termine la preuve. ■

Lemme 3.19. Soient $\Phi(v) \geq 1$ et $\tilde{\alpha}$ défini dans (3.33), alors on a

$$f(\tilde{\alpha}) \leq -\frac{\sqrt{2}}{288(13p+9)}[\Phi(v)]^{\frac{p}{2(p+1)}}. \quad (3.34)$$

Démonstration. En utilisant le Lemme 3.18 avec $\alpha = \tilde{\alpha}$ et de (3.33), nous avons

$$\begin{aligned} f(\tilde{\alpha}) &\leq -\tilde{\alpha}\delta^2 \\ &= \frac{-\delta^2}{(9 + 4\pi p)(8\delta + 2)^{\frac{p+2}{p+1}}} \\ &\leq \frac{-\delta^2}{(9 + 4\pi p)(8\delta + 2(2\delta))^{\frac{p+2}{p+1}}} \\ &= \frac{-\delta^2\delta^{-\frac{(p+2)}{p+1}}}{(9 + 4p\pi)12^{\frac{p+2}{p+1}}}, \end{aligned}$$

ainsi

$$\frac{p+2}{p+1} = 1 + \frac{1}{p+1} \leq 2, \forall p \geq 2.$$

Donc

$$f(\tilde{\alpha}) \leq \frac{-\delta^{\frac{p}{p+1}}}{144(13p+9)},$$

d'après (3.31), nous avons

$$f(\tilde{\alpha}) \leq \frac{-\sqrt{2}}{288(13p+9)} [\Phi(v)]^{\frac{p}{2(p+1)}}.$$

Ce qu'il fallait démontrer. ■

Nombre total d'itérations

Nous avons besoin de compter combien d'itération interne nécessaire pour retourner à la situation $\Phi(v) \leq \tau$ après une μ mise à jour. On définit la valeur de $\Phi(v)$ après la première mise à jour de μ par $(\Phi)_0$ et on note par $(\Phi)_k, k = \overline{1, K}$, la suite des valeurs dans la même itération externe où K désigne le nombre total d'itérations internes dans une itération externe. Par la diminution de $f(\tilde{\alpha})$ pour $k = \overline{0, k-1}$, on a

$$(\Phi_{k+1}(v)) \leq (\Phi_k(v)) - \frac{\sqrt{2}}{288(13p+9)} [\Phi_k]^{\frac{p}{2(p+1)}}.$$

Lemme 3.20. Soit t_0, t_1, \dots, t_k , une suite des nombres positifs, avec

$$t_{k+1} \leq t_k - \beta t_k^{1-\gamma}, k = \overline{0, k-1}, \beta > 0, 0 \leq \gamma \leq 1,$$

alors

$$k \leq \frac{t_0^\gamma}{\beta\gamma}.$$

Si on prend $t_k = (\Phi(v))_k$, $\gamma = 1 - \frac{p}{2(p+1)} = \frac{p+2}{2(p+1)}$ et $\beta = \frac{\sqrt{2}}{288(13p+9)}$. Le nombre total d'itérations internes est borné par

$$k \leq \left(\frac{288(13p+9)(p+1)\sqrt{2}}{(p+2)} \right) (\Phi)_0^{\frac{p+2}{2(p+1)}}$$

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

Donc, le nombre total d'itérations pour avoir une solution approchée avec $n\mu < \epsilon$ est majoré par

$$\left(\frac{288(13p+9)(p+1)\sqrt{2}}{(p+2)} \right) (\Phi)_0^{\frac{p+2}{2(p+1)}} \frac{\log \frac{n}{\epsilon}}{\theta}. \quad (3.35)$$

Pour les méthodes à grand-pas avec $\tau = O(n)$ et $\theta = \Theta(1)$, on a

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

Dans le cas d'une méthode à petit-pas, on a $\tau = O(1)$ et $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$. Le remplacement de ces valeurs dans (3.35), nous donne la meilleure limite possible. Une meilleure borne est obtenue comme suit

D'après (3.25), on a

$$\Psi(t) \leq \left(1 + \frac{\pi}{8}p\right)(t-1)^2, t > 1.$$

Ce qui donne

$$\begin{aligned} \Phi(v_+) &\leq n\Psi\left(\frac{1}{\sqrt{1-\theta}}\sigma\left(\frac{\Phi(v)}{n}\right)\right) \\ &\leq n\left(1 + \frac{\pi}{8}p\right)\left(\frac{\sigma\left(\frac{\Phi(v)}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2 \\ &= \frac{n\left(1 + \frac{\pi}{8}p\right)}{(1-\theta)}\left(\sigma\left(\frac{\Phi(v)}{n}\right) - \sqrt{1-\theta}\right)^2. \end{aligned}$$

De $\sigma(s) \leq 1 + \sqrt{2s}$, on trouve

$$\Phi(v_+) \leq \frac{n\left(1 + \frac{\pi}{8}p\right)}{1-\theta} \left(1 + \sqrt{\frac{2\Phi(v)}{n}} - \sqrt{1-\theta}\right)^2.$$

Et comme $\Phi(v) \leq \tau$, on trouve

$$\begin{aligned} \Phi(v_+) &\leq \frac{n\left(1 + \frac{\pi}{8}p\right)}{(1-\theta)} \left(\theta + \sqrt{2\frac{\tau}{n}}\right)^2 \\ &= \frac{n\left(1 + \frac{\pi}{8}p\right)}{(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2 \\ &= (\Phi)_0. \end{aligned}$$

Nous avons également utilisé

$$1 - \sqrt{1 - \theta} = \frac{\theta}{1 + \sqrt{1 - \theta}} \leq \theta.$$

En utilisant la borne supérieure $(\Phi)_0$, nous obtenons un majorant de nombre total d'itérations suivant

$$\left(\frac{288(13p + 9)(p + 1)\sqrt{2}}{(p + 2)} \right) (\Phi)_0^{\frac{p+2}{2(p+1)}} \frac{\log \frac{n}{\epsilon}}{\theta}.$$

Pour $(\Phi)_0 = O(p)$ la complexité algorithmique devient

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

Conclusion

Dans ce travail, on a résout le problème de programmation linéaire, noté (P) par une méthode de point intérieur (MPI) de type trajectoire centrale (TC), puis on a introduit la notion des fonctions noyaux, où on a présenté une nouvelle fonction noyau à terme barrière trigonométrique (3.11). Cette dernière est proposée par Bouafia et autres [2] pour trouver une classe de direction dans le but d'améliorer la complexité algorithmique des méthodes de (MPI) de type (TC) primal-dual pour les deux versions à grand- et à petit-pas. Les résultats obtenues par les auteurs sont des contributions importantes pour la complexité algorithmique du problème étudié (P).

Bibliographie

- [1] **Bai, Y.Q., El Ghami, M., Roos, C.** : A comparative study of kernel functions for primal-dual interior point algorithms in linear optimization. *SIAM. J. Optim.* 15(1), 101–128 (2004)
- [2] **Bouafia, M., Benterki, D., Yassine, A.** : An Efficient Primal-Dual Interior Point Methode For Linear Programing Based on a New Kernel Function with a Trigonometric Barrier Term (2016).
- [3] **Cai, X.Z., Wang, G.Q., El Ghami, M., Yue, Y.J.** : Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term. *Abstr. Appl. Anal.*, Art. ID 710158, 11 (2014).
- [4] **Cho, G.M.** : An interior point algorithm for linear optimization based on a new barrier function. *Appl. Math. Comput.* 218, 386–395 (2011).
- [5] **ElGhami, M., Guennoun, Z.A., Bouali ,S., Steihaug,T.** :Interior point method for linear optimization based on a kernel function with a trigonometric barrier term. *J. Comput. Appl. Math.* 236, 3613–3623 (2012).
- [6] **Karmarkar, N.K.** : A new polynomial-time algorithm for linear programming. In : *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, vol. 4, pp. 373–395 (1984).
- [7] **Menniche, L.** : Etude théorique et numérique d’une classe de méthodes de points intérieurs pour la programmation linéaire, Thèse de doctorat en sciences, Université Ferhat abbas, Sétif-1, (2017).
- [8] **Peng, J., Roos,C.,Terlaky, T.** : Self-Regularity : A New Paradigm for Primal-Dual Interior Point Algorithms. *Princeton University Press, Princeton* (2002).
- [9] **Peyghami, M.R., Hafshejani, S.F., Shirvani, L.** : Complexity of interior point methods for linear optimization based on a new trigonometric kernel function. *J. Comput. Appl. Math.* 255, 74–85 (2014).

-
- [10] **Peyghami, M.R., Hafshejani, S.F.** : Complexity analysis of an interior point algorithm for linear optimization based on a new proximity function. *Numer. Algorithms* 67, 33–48 (2014).
- [11] **Touil, I.** : Étude comparatives des performances d'une méthode de points intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire, Mémoire de magistère, Université Ferhat abbas, (2005).
- [12] **Roumimili, H.** : Étude qualitative des méthodes de points intérieurs non réalisable pour la (*PL*) et la (*PQC*) (1998).