

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

Ministère de l'enseignement Supérieur et de la Recherche Scientifique

**UNIVERSITE Mohamed Seddik Ben Yahia – Jijel**

**Faculté des Sciences Exactes et Informatique**

**Département de Mathématiques**



## **Mémoire**

Pour l'obtention du diplôme de : **Master**

**Spécialité : Mathématiques Appliquées**

**Option : E.D.P et applications**

**Thème**

**La méthode de Points intérieurs Primale-Duale Pour  
La Programmation Linéaire**

**Présenté par :**

**Boudib Mounira et Laib Sihem**

**Devant le jury :**

**Président: W.Khellaf**

**M.C.B Université de Jijel**

**Encadreur: I.Touil**

**M.C.B Université de Jijel**

**Examineur: L.Menniche**

**M.C.B Université de Jijel**

**Promotion 2016/2017**



# Remerciements

*Au terme de ce modeste travail, nous remercions dieu le  
tout puissant de nous avoir donné le courage et  
la patience afin de réaliser ce travail.*

*Nous tenons tous particulièrement adresses  
nos remerciement les plus vifs d'abord*

*à notre encadreur « **Touil Imene** » qui nous fait*

*l'honneur de dirigé ce travail et nous guidés tout le long  
de son élaboration, nous lui somme très reconnaissant  
pour ses conseils et sa disponibilité.*

*Nos vifs remerciements aux enseignants du département  
de mathématique.*

*Nous remercions Les membres de jury « **W.Khellaf** » et « **L.Mennich** »*

*Pour l'honneur qu'ils nous on fais en acceptant de siéger*

*Parmi notre jury du mémoire.*

*Nous remercions nos familles pour leur patience et soutien et*

*Sans oublier toutes les personnes qui sont participés  
de prés où de loin à l'élaboration de ce mémoire sur tout*

*« **Mr Kecies Mohammed** » enseignant à l'univesité des Mila.*

***merci***

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement Supérieur et de la Recherche Scientifique  
Université de Jijel  
Faculté des Sciences Exactes et Informatique  
Département de mathématiques

N d'ordre :  
N de série :

## Mémoire

Pour l'obtention du diplôme de : **Master**  
**Spécialité** : Mathématiques  
**Option** : Mathématiques Appliquées

---

---

# La méthode de points intérieurs primale-Duale pour la programmation linéaire

---

---

**Présenté par :**

-Boudib Mounira  
-Laib Sihem

**Devant le jury :**

<b>Président :</b>	M.C.B	W. KHELLAF
<b>Encadreur :</b>	M.C.B	I. TOUIL
<b>Examineur :</b>	M.C.B	L. MENNICHE

Promotion 2016-2017

# Table des matières

<b>Introduction Générale</b>	<b>4</b>
<b>1 Préliminaires et notions fondamentales</b>	<b>6</b>
1.1 Analyse convexe . . . . .	6
1.1.1 Notions de convexité . . . . .	6
1.1.2 Convexité et dérivée . . . . .	8
1.1.3 Fonction barrière . . . . .	8
1.2 Programmation mathématique . . . . .	9
1.2.1 Définitions . . . . .	9
1.2.2 Classification des problèmes mathématiques . . . . .	10
1.2.3 Principaux résultats d'existence . . . . .	11
1.2.4 Conditions d'optimalité . . . . .	11
1.3 Programmation linéaire(PL) . . . . .	11
1.3.1 Les formes usuelles d'un programme linéaire . . . . .	12
1.3.2 Relation entre forme standard et forme canonique . . . . .	13
1.3.3 La dualité en programmation linéaire . . . . .	14
1.3.4 Relation primal-dual pour la programmation linéaire . . . . .	15
<b>2 Méthodes de résolution d'un programme linéaire</b>	<b>17</b>
2.1 Méthode du simplexe . . . . .	17
2.1.1 L'algorithme primal du simplexe . . . . .	18
2.1.2 Caractéristiques de l'algorithme du simplexe . . . . .	18
2.2 Méthodes de points intérieurs . . . . .	19
2.2.1 Méthode affines . . . . .	19
2.2.2 Méthodes de réduction du potentiel . . . . .	20
2.2.3 Méthodes de trajectoire centrale . . . . .	21
2.3 Méthodes de trajectoire centrale (TC) pour la programmation linéaire . . . .	21
2.3.1 Facteur de centralité . . . . .	23
2.3.2 Calcul de la direction . . . . .	23
2.3.3 Calcul du pas de déplacement . . . . .	25

<b>3</b>	<b>Méthode de trajectoire centrale via les fonctions noyaux</b>	<b>27</b>
3.1	Méthode de TC primale-duale . . . . .	27
3.2	Fonctions noyaux et propriétés . . . . .	29
3.2.1	Algorithme prototype de la méthode primale-duale : . . . . .	30
3.2.2	Quelques fonctions noyaux . . . . .	30
3.3	La nouvelle fonction noyau et leur caractérisations . . . . .	31
3.3.1	La borne supérieure de la fonction de proximité après une $\mu$ - mise à jour . . . . .	42
3.3.2	Analyse de la complexité de l'algorithme . . . . .	45
3.3.3	Nombre total d'itérations . . . . .	50
	<b>Conclusion Générale</b>	<b>53</b>
	<b>Bibliographie</b>	<b>53</b>

# Introduction Générale

La programmation linéaire, branche de l'optimisation, s'occupe de la minimisation (ou maximisation) d'une fonction à plusieurs variables sous des contraintes (domaine réalisable). Elle permet de déterminer la meilleure solution d'un problème dont les données et les inconnues satisfont à une série d'équations et d'inéquations linéaires.

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

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

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

Dans notre étude, nous avons pris comme modèle la méthode de trajectoire centrale primale-duale, à cause de ses avantages par rapport aux autres classes de méthodes de points intérieurs.

Nous avons proposé un algorithme contenant des modifications consistantes par l'introduction des nouvelles techniques parmi lesquelles les fonctions noyaux pour trouver une classe de directions dans le but d'améliorer le comportement de l'algorithme (c'est-à-dire la complexité algorithmique).

Le mémoire est organisé et divisé en trois chapitres comme suit :

Dans le premier chapitre, on présente une introduction de certaines notions de base et résultats qui seront utilisés par la suite, à savoir : l'analyse convexe, la programma-

tion mathématique, les principaux résultats d'existence, les conditions d'optimalité et la programmation linéaire.

Le deuxième chapitre contient les méthodes de résolutions d'un problème de programmation linéaire telle que la méthode du simplexe et les méthode de points intérieurs où on présente d'une manière détaillée le principe de ces méthodes,

Le troisième chapitre est consacré à la méthode de trajectoire centrale pour la programmation linéaire via les fonctions noyaux où en introduisant la notion des fonctions noyaux pour trouver une nouvelle classe de directions et analysant la complexité de notre algorithme.

# Préliminaires et notions fondamentales

---

## 1.1 Analyse convexe

### 1.1.1 Notions de convexité

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

est appelée matrice Hessienne de  $f$  en  $a$ .

### 1.1.2 Convexité et dérivée

**Définition 1.1.12** Soit  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction deux fois continûment différentiable sur un domaine convexe  $C$ .

1.  $f$  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.  $f$  est une fonction strictement convexe sur  $C$  si et seulement si la matrice Hessienne est définie positive i.e.,

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

### 1.1.3 Fonction barrière

**Définition 1.1.13** L'intérieur relatif d'un sous ensemble non vide  $D$  de  $\mathbb{R}^n$ , noté  $\text{Int}(D)$  est défini comme l'intérieur de  $D$  ou comme sous ensemble de  $\text{aff}(D)$ .

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

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

- Si  $\text{Int}(D) = D$ , alors  $D$  est dit relativement ouvert.

**Définition 1.1.14** On appelle fonction barrière toute fonction  $f$  qui vérifie :

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

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

## 1.2 Programmation mathématique

### 1.2.1 Définitions

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

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

Autrement dit :

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

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

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

Autrement dit :

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

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

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

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

Où :

1.  $S$  est une partie de  $\mathbb{R}^n$  et  $x$  un vecteur appelé variable, ses  $n$  composantes sont dites les inconnues du problème (PM).

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

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

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

**Définition 1.2.3** (Minimum local)

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

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

L'ensemble des minima locaux de (PM) et noté par :  $\text{loc} \min_D f(x)$ .

**Définition 1.2.4** (Minimum global)

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

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

L'ensemble des minima globaux de (PM) et noté par :  $\text{arg} \min_D f(x)$ .

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

## 1.2.2 Classification des problèmes mathématiques

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

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

### 1.2.3 Principaux résultats d'existence

**Théorème 1.2.1** (Weierstrass)

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

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

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

### 1.2.4 Conditions d'optimalité

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

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

**Théorème 1.2.3** (Karush Kuhn Tucker (KKT))

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

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

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

## 1.3 Programmation linéaire (PL)

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

La représentation mathématique de (PL) est donnée par :

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

Ou bien sous forme matricielle

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

Où (PL) est considéré comme primal.

Tels que

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

### 1.3.1 Les formes usuelles d'un programme linéaire

#### a) La forme canonique

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

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

**Cas de maximisation :**

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

**Cas de minimisation :**

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

**Exemple 1.3.1 Cas de maximisation :**

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

**Cas de minimisation :**

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

#### b) La forme standard

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

1. Toutes ses variables sont non négatives.
2. Toutes les contraintes sous forme des équations.

3. La fonction objectif soit maximisé ou minimisé.
4.  $b_i, i = 1 \dots m$ , sont non négatives.

**Cas de maximisation :**

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

**Cas de minimisation :**

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

**Exemple 1.3.2 Cas de maximisation :**

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

**Cas de minimisation :**

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

### c) La forme mixte

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

**Exemple 1.3.3**

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

## 1.3.2 Relation entre forme standard et forme canonique

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

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

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

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

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

a)  $Ax \leq b \Leftrightarrow \{Ax + e = b\}$ , avec  $e \geq 0$ .

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

### 1.3.3 La 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 par (D)).

#### Dual d'un programme linéaire

Soit le programme linéaire primal écrit sous forme standard

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

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

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

**Preuve.** On considère la fonction lagrangienne suivante

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

D'où

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

Donc le problème dual de (p) est

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

■

**Exemple 1.3.4** Soit le programme primal sous forme standard

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

Son dual est

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

### 1.3.4 Relation primal-dual pour la programmation linéaire

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

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

#### Remarque 1.3.1

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

#### Théorème 1.3.1 (dualité faible)

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

**Preuve.** Or  $x$  et  $y$  sont des solutions réalisables, alors

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

■

**Théorème 1.3.2** (*dualité forte*)

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

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

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

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

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

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

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

D'où le resultat. ■

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

2. Le programme primal a une solution optimale si et seulement si le programme dual a aussi une.

Pour plus amples de détails sur certains résultats de ce chapitre, le lecteur peut voir par exemple [1, 8, 9, 10, 6].

Dans le chapitre suivant, on présente des méthodes numériques pour résoudre un problème de la programmation linéaire, en proposant des méthodes typiques, telle que la méthode du simplexe [8, 6], et celles de points intérieurs [8, 9, 10].

# Méthodes de résolution d'un programme linéaire

---

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

## 2.1 Méthode du simplexe

Elle a été développée à la fin des années 40 par **G.Dantzig**, elle tient compte systématiquement des résultats établis précédemment. Elle évolue sur la frontière du domaine réalisable de sommet en sommet adjacent, en réduisant la valeur de l'objectif jusqu'à l'optimum. Un critère d'optimalité simple permet de reconnaître le sommet optimal, le nombre de sommet étant fini, l'algorithme ainsi défini converge en un nombre fini d'itérations n'excédent pas le nombre  $c_n^m = \frac{n!}{m!(n-m)!}$  sous l'hypothèse que tous les sommets visités sont non dégénérés.

Dans le cas dégénéré l'algorithme risque de cycler, cependant il existe des techniques convenables pour éviter ce phénomène. En général, la méthode de simplexe possède un comportement numérique très satisfaisant confirmé par ses applications multiples dans la résolution d'une large classe de problèmes pratiques.

En théorie, la méthode n'a pas autant de succès, elle est plutôt jugée inefficace par sa complexité arithmétique exponentielle qui est d'ordre  $O(2^n)$  opérations voir [1].

### 2.1.1 L'algorithme primal du simplexe

On suppose que l'on dispose d'une base réalisable de départ  $B^0$ . Les différentes étapes de l'algorithme sont les suivantes voir [6]

1.  $B^0$  base réalisable de départ  $x^0 = [x_B^0, x_N^0] = [B^{-1} \cdot b, 0]$ . Itération  $k = 0$ .
2.  $k \leftarrow k + 1$ .
3. à l'itération  $k$  soit  $B$  la base courante,  $x = [x_B, x_N]$  la solution de base correspondante où  $x_B = B^{-1} \cdot b - B^{-1} \cdot N \cdot x_N$  et  $x_N = x_N^0 + \theta \cdot e_s$  ( $e_s$  est un vecteur de même dimension que  $x_N$  avec toutes ses composantes nulles sauf la composante  $s$  égale à 1)  $\theta$  doit être choisi de telle sorte que  $x = [x_B, x_N]$  reste une solution c'est-à-dire

$$\begin{cases} x_B = B^{-1} \cdot b - B^{-1} \cdot N \cdot x_N = B^{-1} \cdot b - \theta B^{-1} \cdot N \cdot e_s, \\ x_B \geq 0. \end{cases}$$

Calculer :

- (a)  $\bar{b} = B^{-1} \cdot b$ , (les valeurs des variables de base)
- (b)  $\pi = c_B \cdot B^{-1}$ , (les multiplicateurs du simplexe)
- (c)  $\bar{c}_N = c_N - \pi \cdot N$ . (les coûts réduits)
4. Si  $\bar{c}_N \geq 0$ , STOP : l'optimum est atteint.  
Si  $\exists s$  tel que :  $\bar{c}_s < 0$  alors :
5. Soit  $A_s$  la colonne  $s$  de  $A$ .  
Calculer  $\bar{A}_s = B^{-1} \cdot A_s$ .  
Si  $\bar{a}_{is} \leq 0, \forall i = 1, \dots, m$ , STOP : optimum non borné  $(-\infty)$ .  
Sinon, calculer :  
 $\widehat{x}_s = \frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{i/\bar{a}_{is} > 0} \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} \right\}$ .
6. Soit  $x_t$  la variable correspondant à la  $r$ -ième ligne de la base c'est-à-dire telle que  $B^{-1} \cdot A_t = e_r$  ( $m$ -vecteur à composantes toutes nulles sauf la composante  $r$  égale à +1); alors la variable  $s$  prend la valeur  $\widehat{x}_s > 0$  (rentre en base); la variable  $t$  s'annule ( $\widehat{x}_t = 0$ ) (sort de la base); la nouvelle solution courante  $\widehat{x}$  correspond à la nouvelle base réalisable :  $\widehat{B} = B + \{A_s\} - \{A_t\}$ . Calculer l'inverse de la nouvelle base  $\widehat{B}^{-1}$  et retourner en (2).

### 2.1.2 Caractéristiques de l'algorithme du simplexe

On peut dégager les remarques et les propriétés suivantes voir [8]

1. Cet algorithme nécessite un point de départ réalisable.
2. Cet algorithme ne permet pas de traiter naturellement les variables libres, donc il faut décomposer ces variables ce qui augmente la taille du vecteur  $x$  et de la matrice  $A$ .

3. Cet algorithme est sensible à la dégénérescence (problème de cyclage).
4. Bien que très efficace en pratique, cet algorithme ne satisfait pas les théoriciens à cause de sa complexité exponentielle (le nombre d'itérations nécessaire pour atteindre la solution optimale est exponentiel en fonction de la taille du problème ( $n$ )).

## 2.2 Méthodes de points intérieurs

Les méthodes de points intérieurs comme leur nom l'indique évitent soigneusement la frontière de l'ensemble admissible et donc ne souffriront pas de l'aspect combinatoire inhérent à la méthode du simplexe.

Ces méthodes ont été développées dans les années 60 dans le but de résoudre des programmes mathématiques non linéaires. Leur utilisation pour la programmation linéaire n'a pas reçu autant d'enthousiasme à cause de la dominance quasi-totale de la méthode du simplexe à cette époque.

Après l'apparition de l'algorithme de Karmarkar en 1984 pour la programmation linéaire, les méthodes de points intérieurs ont connu une véritable révolution, elles sont devenues compétitives à la méthode du simplexe surtout pour les problèmes de grande taille (plus de mille variables ou contraintes).

On distingue trois classes fondamentales de méthodes de points intérieurs à savoir

1. Les méthodes affines.
2. Les méthodes de réduction du potentiel (noté RP).
3. Les méthodes de trajectoire centrale (chemin central) (noté TC).

### 2.2.1 Méthode affines

Il s'agit pratiquement de l'algorithme de Karmarkar sans fonction potentiel et sans transformation projective, on utilise une transformation affine et on remplace la contrainte de non négativité par un ellipsoïde qui contient le nouvel itéré.

L'algorithme est d'une structure simple, malheureusement, il n'est pas facile de démontrer sa complexité polynomiale voir [1, 8, 9].

#### Propriétés et commentaires

1. C'est une méthode primale, performante en pratique, quoique sensible aux choix initial.

2. Elle est simple au sens qu'elle ne demande ni transformation projective au fonction potentiel, ni même la préparation du problème (forme canonique).
3. La démonstration de la convergence est relativement simple dans le cas non dégénéré, elle est fastidieuse en cas d'échéant.
4. Il n'y a pas de résultat de polynomialité on pense plutôt (pour de très bonne raisons) que l'algorithme est d'une complexité exponentielle.

C'est sans doute la raison pour laquelle il a resté inconnu avant les investigations de Karmarkar.

## 2.2.2 Méthodes de réduction du potentiel

Ces méthodes son typiquement désignées pour trouver les solutions améliorantes du programme non linéaire suivant

$$\begin{cases} \min \left( f(x, y, s) = q \ln(c^t x - b^t y) - \sum_{j=1}^n \ln(x_j) \right) \\ Ax = b, x \geq 0, \\ A^t y + s = c, s \geq 0. \end{cases}$$

Où la fonction objective  $f$  est appelée fonction potentielle et  $q$  son paramètre. C'est ce type de fonction que Karmarkar introduit dans son article avec  $q = 1$ .

Les méthodes de réduction du potentiel ont pour principe d'imposer la réduction d'une fonction potentiel non linéaire à chaque étape, tout en maintenant une certaine distance avec la frontière du domaine réalisable [7].

L'algorithme de Karmarkar de 1984 appartient à cette catégorie.

### Propriétés et commentaires

1. Loin d'être aussi simple que les méthodes affines.
2. Plus attractive pour au moins deux raisons :
  - a Leur complexité polynomiale.
  - b Leur compatibilité avec les techniques de recherche linéaire sous le pas de déplacement ce qui conduit à un comportement numérique rivalisant.
3. Elles sont primales-duales.
4. Les transformations projectives ne sont pas nécessaires ni en théorie ni en pratique.
5. Elles n'ont pas reçu beaucoup d'attention au niveau des simulations numériques peut être à cause des difficultés algorithmiques rencontrées au début mais aussi leur nature de premier ordre.

### 2.2.3 Méthodes de trajectoire centrale

Elles 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 : la complexité polynomiale et la convergence super linéaire. Les algorithmes de trajectoire centrale restreintes les itérés à un voisinage de la trajectoire centrale, ce dernier est une courbe de points strictement réalisables [9, 10].

Dans ce mémoire on s'intéresse à une méthode de trajectoire centrale de type primale-duale que l'on décrit dans la section suivante.

## 2.3 Méthodes de trajectoire centrale (TC) pour la programmation linéaire

L'idée générale de cette méthode consiste à suivre un chemin particulier (chemin de centres) constitué d'une suite de points intérieurs en prenant comme direction de déplacement celle de Newton et un pas de déplacement convenable.

Soit le programme non linéaire paramétrisé associé à  $(P)$

$$(P_\mu) \begin{cases} \min \left[ c^t x - \mu \sum_{j=1}^n \ln(x_j) = f_\mu(x) \right] \\ Ax = b, x > 0. \end{cases}$$

et soit  $(D_\mu)$  le programme non linéaire paramétrisé associé à  $(D)$

$$(D_\mu) \begin{cases} \max \left[ b^t y - \mu \sum_{i=1}^n \ln(s_i) = g_\mu(y) \right] \\ A^t y + s = c, s > 0. \end{cases}$$

Où  $\mu > 0$ , désigne le paramètre barrière.

**Lemme 2.3.1** *La fonction  $f_\mu(x)$  est strictement convexe.*

**Preuve.** Pour la preuve, il suffit de montrer que la matrice Hessienne est définie positive.

On a  $f_\mu(x) = c^t x - \mu \sum_{j=1}^n \ln(x_j) \in C^\infty$  et nous avons en particulier  $\nabla f_\mu(x) = c - \mu X^{-1}e$ , avec  $e = (1, \dots, 1)^t \in \mathbb{R}^n$  et  $\nabla^2 f_\mu(x) = \mu X^{-2}$ ,

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

La résolution de  $(P)$  (respectivement  $(D)$ ) est équivalente à celle de  $(P_\mu)$  (respectivement  $(D_\mu)$ ), dans le sens que si  $x^*(\mu)$  (respectivement  $(y^*(\mu), s^*(\mu))$ ) est une solution optimale de

$(P_\mu)$  (respectivement  $(D_\mu)$ ) alors  $x^* = \lim_{\mu \rightarrow 0} x^*(\mu)$  (respectivement  $(y^*, s^*) = \lim_{\mu \rightarrow 0} (y^*(\mu), s^*(\mu))$ ) est une solution optimale de  $(P)$  (respectivement  $(D)$ ).

Donc, il suffit de résoudre  $(P_\mu)$  (respectivement  $(D_\mu)$ ).

Notons les ensembles des solutions strictement réalisable de  $(P)$  et  $(D)$  respectivement par

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

$$T_{int} \{(y, s) \in \mathbb{R}^m \times \mathbb{R}_+^n; A^t y + s = c, s > 0\}.$$

**Remarque 2.3.1** On suppose dans tous le reste de ce chapitre que :

$(H_1)$  :  $S_{int}$  est non vide.

$(H_2)$  :  $T_{int}$  est non vide.

**Proposition 2.3.1** 1. Sous l'hypothèse  $(H_1)$ , on a

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

2. Sous l'hypothèse  $(H_2)$ , on a

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

Comme la fonction  $f_\mu$  est strictement convexe et les contraintes sont linéaires, alors les conditions d'optimalités de **KKT** sont nécessaires et suffisantes pour les deux problèmes  $(P_\mu)$  et  $(D_\mu)$  respectivement

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

On pose  $\mu X^{-1}e = s$ , il vient

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

De même pour  $(D_\mu)$ , on obtient

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

Où

$X = \text{diag}(x_i)$ ,  $X^{-1} = \text{diag}(\frac{1}{x_i})$ ,  $S = \text{diag}(s_i)$ ,  $S^{-1} = \text{diag}(\frac{1}{s_i})$ ,  $i = 1, \dots, n$  et  $e = (1, \dots, 1)^t \in \mathbb{R}^n$ .

Pour chaque valeur de  $\mu > 0$ , le système  $(CNS(P_\mu))$  admet une solution unique  $(x(\mu), y(\mu), s(\mu))$ . Cette solution s'appelle  $\mu$ -centre de  $(P)$  et de  $(D)$ .

L'ensemble des  $\mu$ -centres,  $T_c = \{(x(\mu), y(\mu), s(\mu)); \mu > 0\}$ , construit la trajectoire centrale qui converge vers la solution optimale primale-duale de problèmes  $(P)$  et  $(D)$  lorsque  $\mu$  tend vers 0.

### 2.3.1 Facteur de centralité

Pour mesurer la qualité de la solution trouvée on introduit un facteur dit de "centralité" ou de "proximité" défini par le scalaire  $\delta(x, s, \mu) = \left\| \frac{s(\mu)x(\mu)}{\mu} - e \right\|$ .

La trajectoire centrale est alors l'ensemble des points réalisables de proximité nulle pour un certain  $\mu$ .

On dit que le triplet  $(x(\mu), y(\mu), s(\mu))$  est proche de la trajectoire centrale (voisin de la trajectoire centrale), s'il appartient à l'ensemble

$$S_{cent}(\sigma) = \{(x, y, s) \in S_{int} \times T_{int}; \|s(\mu)x(\mu) - \mu e\| \leq \sigma\mu; 0 < \sigma < 1\}.$$

– Si  $\mu \rightarrow 0$ , alors  $(x(\mu), y(\mu), s(\mu))$  converge vers la solution optimale de  $(P)$  et  $(D)$ , ce qui donne le Saut de dualité nul, i.e.,  $x^t s = c^t x - b^t y = \mu x^t X^{-1} e = n\mu = 0$ .

Ceci veut dire que  $(TC)$  se termine en une solution optimale du problème primal-dual.

### 2.3.2 Calcul de la direction

Puisque le système (2.1) est non linéaire, une résolution directe est en général impossible. Alors on se limite à chercher des solutions approchées suivant la trajectoire centrale en formant une suite décroissante des sauts de dualité qui tend vers 0 à la limite.

La méthode de Newton est considérée dans ce cas comme l'une des meilleures méthodes de résolution.

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

$$F(x, y, s) = \begin{pmatrix} Ax - b \\ A^t y + s - c \\ XSe - \mu e \end{pmatrix}; x, s > 0.$$

La direction de Newton  $\Delta w = (\Delta x, \Delta y, \Delta s)$  est la solution du système linéaire

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

telle que  $DF(x, y, s)$  est la matrice Jacobienne de  $F$  au point  $(x, y, s)$  donnée par

$$DF(x, y, s) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ S & 0 & X \end{pmatrix}.$$

Où  $I = I_n$  est la matrice d'identité d'ordre  $n$ .

On résout le système (2.2) on trouve

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu e - XSe \end{pmatrix}. \quad (2.3)$$

Ce qui donne

$$\begin{cases} 1) A\Delta x = 0, \\ 2) A^t\Delta y + \Delta s = 0, \\ 3) S\Delta x + X\Delta s = \mu e - XSe, \end{cases}$$

de (2), on a

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

On remplace  $\Delta s$  dans (3), on trouve

$$S\Delta x - XA^t\Delta y = \mu e - XSe.$$

Alors

$$S\Delta x = XA^t\Delta y + \mu e - XSe,$$

d'où

$$\Delta x = S^{-1}(XA^t\Delta y + \mu e - XSe).$$

On remplace  $\Delta x$  dans (1), on obtient

$$A(S^{-1}(XA^t\Delta y + \mu e - XSe)) = 0,$$

Donc

$$AS^{-1}XA^t\Delta y + AS^{-1}(\mu e - XSe) = 0.$$

Alors

$$AS^{-1}XA^t\Delta y = -AS^{-1}(\mu e - XSe),$$

d'où

$$\begin{aligned} \Delta y &= -(AS^{-1}XA^t)^{-1}AS^{-1}(\mu e - XSe) \\ &= (AS^{-1}XA^t)^{-1}AS^{-1}(XSe - \mu e). \end{aligned}$$

On trouve

$$\begin{aligned} \Delta x &= -S^{-1}[(XA^t((AS^{-1}XA^t)^{-1}AS^{-1}(\mu e - XSe) + \mu e - XSe)] \\ &= -S^{-1}XA^t(AS^{-1}XA^t)^{-1}AS^{-1}(\mu e - XSe) - S^{-1}(\mu e - XSe) \\ &= (S^{-1}XA^t(AS^{-1}XA^t)^{-1}AS^{-1} + S^{-1})(XSe - \mu e), \end{aligned}$$

et

$$\Delta s = A^t(AS^{-1}XA^t)^{-1}AS^{-1}(\mu e - XSe).$$

D'où

$$\begin{cases} \Delta x = (S^{-1}XA^t(AS^{-1}XA^t)^{-1}AS^{-1} + S^{-1})(XSe - \mu e) \\ \Delta y = (AS^{-1}XA^t)^{-1}AS^{-1}(XSe - \mu e) \\ \Delta s = -A^t(AS^{-1}XA^t)^{-1}AS^{-1}(XSe - \mu e). \end{cases}$$

Le nouvel itéré s'écrit alors sous la forme

$$(x^+, y^+, s^+) = (x, y, s) + (\Delta x, \Delta y, \Delta s).$$

Dans la partie suivante, pour assurer la stricte faisabilité de  $(x^+, y^+, s^+)$  on introduit un paramètre  $\alpha > 0$  appelé le pas de déplacement. Dans ce cas l'itéré de Newton modifié s'écrit

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

### 2.3.3 Calcul du pas de déplacement

Théoriquement la méthode (TC) converge pour tout  $0 < \alpha < 1$ , mais on sait que le pas de déplacement  $\alpha$  et le nombre d'itérations sont inversement proportionnels : « plus  $\alpha$  est grand plus le nombre d'itérations est petit » [7].

Le pas de déplacement  $\alpha$  est choisi d'une façon à maintenir la stricte faisabilité des nouveaux itérés, alors on va choisir une procédure dite de positivité qui est moins coûteuse que la méthode de recherche linéaire.

#### Procédure de positivité :

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

$$x^+ = x + \alpha\Delta x > 0, \tag{2.4}$$

$$s^+ = s + \alpha\Delta s > 0, \tag{2.5}$$

de (2.4), on a

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

Ce qui est équivalent à

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

tel que

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

De même pour (2.5), on a

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

Ce qui est équivalent à

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

tel que

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

Alors, il suffit de prendre le pas de déplacement suivant

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

### Algorithme prototype de la méthode de trajectoire centrale

#### Début algorithme

**Données :** soit  $\epsilon > 0$  un paramètre de précision.

**Initialisation :**  $(x^0, y^0, s^0) \in S_{cent}(\sigma)$  où  $\mu^0 = \frac{(x^0)s^0}{n}, k = 0$ .

#### Etape 01 :

- **Tant que**  $(x^k)^t s^k \geq \epsilon$  faire

#### Etape 02 :

- **Prendre**  $\mu^{k+1} = \mu^k \sigma, 0 < \sigma < 1;$

#### Etape 03 :

Calculer la direction  $\Delta w = (\Delta x, \Delta y, \Delta s)$ , en résolvant le système linéaire (2.3).

#### Etape 04 :

Trouver  $\alpha_1$  et  $\alpha_2$  solutions des inégalités (1.4) et (1.5) respectivement et prendre  $\alpha = \rho \min(\alpha_1, \alpha_2), 0 < \rho < 1$ .

#### Etape 05 :

**Prendre :**  $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha(\Delta x^k, \Delta y^k, \Delta s^k)$ .

#### Etape 06 :

Poser  $k = k + 1$  et aller à l'étape 01.

- **Fin tant que**

**Fin algorithme**

# Méthode de trajectoire centrale via les fonctions noyaux

---

Dans ce chapitre, on va étudier la méthode de trajectoire centrale via les fonctions noyaux. A cet effet, on a choisi de détailler l'article de **B.Kheirfam** et **M.Moslemi**, intitulé " *A comparative study of karnel function for primal-dual interior point algorithm in linear optimisation*" [5], où les auteurs ont introduit une fonction noyau de type trigonométrique, pour trouver une nouvelle classe de direction dans le but d'obtenir un nouveau résultat de complexité.

On considère le problème primal de programmation linéaire suivant

$$(P) \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0. \end{cases} \quad (3.1)$$

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

Son dual est

$$(D) \begin{cases} \max b^t y \\ A^t y + s = c \\ y \in \mathbb{R}^m, s \in \mathbb{R}_+^n. \end{cases} \quad (3.2)$$

## 3.1 Méthode de TC primale-duale

Supposons que les deux problèmes (P) et (D) vérifient la condition de point intérieur (CPI), c'est-à-dire, il existe  $(x^0, y^0, s^0)$  strictement réalisable tel que

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

La *CPI* assure l'existence d'une solution optimale primale-duale  $(x^*, y^*, s^*)$  avec un saut de dualité nul, i.e.,  $c^t x^* - b^t y^* = (x^*)^t s^* = 0$ .

Il est bien connu que trouver une solution optimale de  $(P)$  et  $(D)$  est équivalent à résoudre le système non linéaire suivant

$$\begin{cases} Ax = b, \\ A^t y + s = c, \\ xs = 0, \\ x > 0, s > 0. \end{cases} \quad (3.4)$$

L'idée principale des méthodes de points intérieurs (*MPIs*) de type primal-dual est de remplacer la troisième équation du système (3.4) (Condition de complémentarité) par l'équation paramétrisée  $xs = \mu e$  avec  $\mu > 0$  fixé, donc on obtient le système non linéaire suivant

$$\begin{cases} Ax = b, \\ A^t y + s = c, \\ xs = \mu e, \\ x > 0, s > 0, \mu > 0. \end{cases} \quad (3.5)$$

Ce système à une solution unique  $(x(\mu), y(\mu), s(\mu))$  pour chaque  $\mu > 0$  fixé, qui est appelé  $\mu$ -centre de  $(P)$  et  $(D)$ .

L'ensemble des  $\mu$ -centres donne une courbe paramétrisée, cette dernière est appelée la trajectoire centrale (le chemin central) de  $(P)$  et  $(D)$ .

Si  $\mu$  tend vers 0, alors la limite de ce chemin central existe et, puisque le point de la limite satisfait la condition de complémentarité, la limite produit des solutions optimales pour  $(P)$  et  $(D)$ .

Soit  $(\Delta x, \Delta y, \Delta s)$  la solution du système qui est obtenue par l'application de la méthode de Newton

$$\begin{cases} A\Delta x = 0, \\ A^t \Delta y + \Delta s = 0, \\ s\Delta x + x\Delta s = \mu e - xse. \end{cases} \quad (3.6)$$

Dans ce cas l'itéré de Newton s'écrit comme suit

$$(x^+, y^+, s^+) = (x, y, s) + (\Delta x, \Delta y, \Delta s).$$

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

$$\begin{cases} x^+ = x + \alpha \Delta x, \\ y^+ = y + \alpha \Delta y, \\ s^+ = s + \alpha \Delta s. \end{cases} \quad (3.7)$$

Pour tout vecteur  $x > 0$  et  $s > 0$  on définit le vecteur  $v$  comme suit

$$v = \sqrt{\frac{xs}{\mu}}; xs = (x_1 s_1, \dots, x_n s_n). \quad (3.8)$$

On introduit les directions  $dx$  et  $ds$  comme suit

$$dx = \frac{v\Delta x}{x}; ds = \frac{v\Delta s}{s}. \quad (3.9)$$

Par l'utilisation de (3.8) et (3.9), on peut écrire le système (3.6) comme suit

$$\begin{cases} \bar{A}dx = 0, \\ \bar{A}^t dy + \Delta s = 0, \\ dx + ds = V^{-1} - V, \end{cases} \quad (3.10)$$

où

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

L'observation cruciale est que le second membre de la troisième équation du système (3.10) est le gradient négative de la fonction barrière logarithmique  $\Phi_c(v)$ , i.e.,

$$dx + ds = -\nabla\Phi_c(v),$$

où

$$\Phi_c(v) = \sum_{i=1}^n \Psi_c(v_i), \Psi_c(v) = \frac{v^2 - 1}{2} - \log(v), \quad (3.11)$$

avec  $\Psi_c$  est la fonction noyau classique de la fonction logarithmique barrière  $\Phi_c(v)$ .

## 3.2 Fonctions noyaux et propriétés

**Définition 3.2.1** On appelle  $\Psi : \mathbb{R}_{++} \rightarrow \mathbb{R}_+$  une fonction noyau si elle est deux fois différentiable et vérifie les conditions suivantes :

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

Les conditions 1 et 2 montrent que  $\Psi(v)$  est strictement convexe et minimale en  $v = e$ , avec  $\Psi(e) = 0$ . Ce qui donne

$$\Psi(e) = 0 \Leftrightarrow \nabla\Psi(e) = 0 \Leftrightarrow v = e \Leftrightarrow xs = \mu e.$$

Et qu'elle peut s'écrire sous la forme suivante

$$\Psi(t) = \int_1^t \int_1^\xi \Psi''(\tau) d\tau d\xi. \quad (3.12)$$

La troisième condition montre que  $\Psi(v)$  est une fonction barrière.

### 3.2.1 Algorithme prototype de la méthode primale-duale :

#### Début algorithme

#### Données :

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

$\varepsilon > 0$ , un paramètre de précision,

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

#### Initialisation :

$(x^0, s^0)$  et  $\mu^0 = 1$ , telle que  $\Phi(x^0, s^0, \mu^0) \leq \tau$ ;

$x := x^0; s := s^0; \mu := \mu^0$ ;

#### Tant que $n\mu \geq \varepsilon$ faire

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

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

– Résoudre le système (3.18) et utiliser (3.9) pour déterminer  $(\Delta x, \Delta y, \Delta s)$ ;

– Calculer le pas de déplacement  $\alpha$ ;

– Poser :

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

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

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

Fin tant que ;

Fin tant que ;

#### Fin algorithme.

La question importante est comment choisir les paramètres  $\tau, \theta$  et le pas de déplacement  $\alpha$  qui minimisent la complexité de l'algorithme.

### 3.2.2 Quelques fonctions noyaux

On donne dans le tableau suivant les différentes fonctions noyaux proposées par les chercheurs, et leurs complexité algorithmique à petit-pas et grand-pas.

	Fonction noyau	Complexité algorithmique	Complexité algorithmique
$i$	$\Psi_i(t)$	Petit-pas	Grand-pas
1	$\frac{1}{2}(t^2 - 1) - \log t$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$
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	$\frac{t^2-1}{2} + e^{\frac{1}{t-1}} - 1$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log^2 n \log \frac{n}{\varepsilon})$
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	$(m+1)t^2 - (m+2)t + \frac{1}{m}, m > 4$	$O(q^2 \sqrt{n} \log \frac{n}{\varepsilon})$	$O(m^{\frac{2m+1}{2m}} \log \frac{n}{\varepsilon})$
13	$t^2 - 2t + \frac{1}{\sin(\frac{\pi t}{1+t})}, t > 0$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$

TABLE 3.1 – La complexité à grand et petit-pas

### 3.3 La nouvelle fonction noyau et leur caractérisations

Dans ce qui suit, on remplace le second membre de la troisième équation du système (3.10) par  $-\nabla\Phi(v)$ , où  $\Phi$  est la fonction barrière de la nouvelle fonction noyau  $\Psi(t)$ . Cette dernière est de type trigonométrique, elle est proposée par **B. Kheirfam** et **M. Moslemi** en 2015 dans le but d'améliorer la complexité algorithmique.

Soit la fonction de noyau  $\Psi(t)$  définie par

$$\Psi(t) = t^2 - 2t + \frac{1}{\sin(u(t))}, t > 0, u(t) = \frac{\pi t}{1+t}. \quad (3.13)$$

Alors les trois premières dérivées de  $\Psi$  sont données par

$$\Psi'(t) = 2t - 2 - \frac{u'(t) \cos(u(t))}{\sin^2(u(t))}, u'(t) = \frac{\pi}{(1+t)^2}. \quad (3.14)$$

$$\Psi''(t) = 2 + \frac{u'^2(t) \sin^2(u(t)) - u''(t) \cos(u(t)) \sin(u(t)) + 2u'^2(t) \cos(u(t))}{\sin^3(u(t))}, u''(t) = \frac{-2\pi}{(1+t)^3}. \quad (3.15)$$

Et

$$\begin{aligned} \Psi'''(t) = & \frac{1}{\sin^4(u(t))} \left[ -6u'^3(t) \cos^3(u(t)) + 6u''(t)u'(t) \cos^2(u(t)) \sin(u(t)) + 3u''(t)u'(t) \sin^3(u(t)) \right] \\ & + \frac{1}{\sin^4(u(t))} \left[ -5u'^3(t) \cos(u(t)) \sin^2(u(t)) - u'''(t) \cos(u(t)) \sin^2(u(t)) \right]. \end{aligned} \quad (3.17)$$

Le système (3.10) est peut reformuler comme suit

$$\begin{cases} \bar{A} dx = 0, \\ \bar{A}^t dy + \Delta s = 0, \\ dx + ds = -\nabla \Phi(v), \end{cases} \quad (3.18)$$

où  $\Phi(v)$  est la fonction barrière de la fonction noyau définie dans (3.13).

**Lemme 3.3.1** Soit  $\Psi(t)$  la fonction noyau définie dans (3.13), alors on a

1.  $\pi \cos(u(t)) < (1+t) \sin(u(t)) < \pi, t > 0$ .
2.  $\pi t \cos(u(t)) < (1+t) \sin(u(t)) < \pi t, t > 0$ .
3.  $(1+t) \sin(u(t)) < 2\pi t \cos(u(t)), 0 < t \leq \frac{1}{2}$ .

**Preuve.**

1) Pour  $t > 0$ , on définit les deux fonctions  $f$  et  $g$  comme suit

$$\begin{cases} f(x) = \frac{\pi x}{1+x} - \sin\left(\frac{\pi}{1+x}\right), \\ g(x) = \frac{\pi x}{1+x} + \tan\left(\frac{\pi}{1+x}\right). \end{cases}$$

Alors on a

$$\begin{cases} f'(x) = \frac{\pi x}{(1+x)^2} + \frac{\pi x}{(1+x)^2} \cos\left(\frac{\pi}{1+x}\right) = \frac{\pi x}{(1+x)^2} (1 + \cos\left(\frac{\pi}{1+x}\right)), \\ g'(x) = -\frac{\pi}{(1+x)^2} \tan^2\left(\frac{\pi}{1+x}\right). \end{cases}$$

et comme

$$-1 < \cos(u(t)) < 1, \forall t > 0.$$

On obtient

$$f'(x) > 0.$$

D'où  $f$  est une fonction croissante et comme  $f(0) = 0$ , on trouve  $f(x) > 0$ , ce qui donne d'une part

$$\frac{\pi x}{1+x} > \sin\left(\frac{\pi}{1+x}\right).$$

D'autre part, on a

$$g'(x) < 0.$$

Ce qui implique que  $g$  est une fonction décroissante et comme  $g(0) = 0$ , alors  $g(x) < 0$ .

D'où

$$\frac{\pi x}{1+x} < -\tan\left(\frac{\pi}{1+x}\right),$$

Ce qui donne pour  $x = \frac{1}{t}$

$$\begin{cases} (1+t) \sin(u(t)) < \pi, \\ -\pi \cos(u(t)) < (1+t) \sin(u(t)). \end{cases}$$

D'où le premier résultat

$$\pi t \cos(u(t)) < (1+t) \sin(u(t)) < \pi t, t > 0.$$

2) Pour montrer le deuxième résultat, on définit les fonctions  $f$  et  $g$  comme suit

$$\begin{cases} f(t) = \frac{\pi t}{1+t} - \sin\left(\frac{\pi t}{1+t}\right), \\ g(t) = \frac{\pi t}{1+t} - \tan\left(\frac{\pi t}{1+t}\right). \end{cases}$$

En suivant les mêmes étapes du premier résultat, on peut voir facilement que  $f(t)$  est une fonction croissante et que  $g(t)$  est une fonction décroissante et que  $f(0) = g(0) = 0$ . Par conséquent  $f(t) > 0$  et  $g(t) < 0$ . Ce qui implique

$$\pi t \cos(u(t)) < (1+t) \sin(u(t)) < \pi t, \forall t > 0.$$

3) Pour  $0 < t \leq \frac{1}{2}$ , on définit la fonction  $f$  comme suit

$$f(t) = 2\pi t \cos\left(\frac{\pi t}{1+t}\right) - (1+t) \sin\left(\frac{\pi t}{1+t}\right),$$

Alors on a

$$\begin{aligned} f'(t) &= 2\pi \cos\left(\frac{\pi t}{1+t}\right) - 2\pi t \frac{\pi t}{(1+t)^2} \sin\left(\frac{\pi t}{1+t}\right) - (1+t) \frac{\pi}{(1+t)^2} \cos\left(\frac{\pi t}{1+t}\right) - \sin\left(\frac{\pi t}{1+t}\right) \\ &= \left(2\pi - \frac{\pi}{1+t}\right) \cos\left(\frac{\pi t}{1+t}\right) - \left(\frac{2\pi^2 t}{(1+t)^2} + 1\right) \sin\left(\frac{\pi t}{1+t}\right), \end{aligned}$$

et

$$f''(t) = -\frac{2\pi^2}{(1+t)^2} \sin\left(\frac{\pi t}{1+t}\right) - \frac{\pi^2(1-2t)}{(1+t)^3} - \frac{2\pi^3 t}{(1+t)^4} \cos\left(\frac{\pi t}{1+t}\right) < 0.$$

Puisque pour  $0 < t \leq \frac{1}{2}$ , on a

$$\sin\left(\frac{\pi t}{1+t}\right) > 0, \cos\left(\frac{\pi t}{1+t}\right) > 0.$$

Ce qui implique que pour  $0 < t \leq \frac{1}{2}$  la fonction  $f(t)$  est strictement concave, puisque  $f(0) > 0$  et  $f(\frac{1}{2}) > 0$ , par conséquent  $f(t) > 0$ , donc

$$2\pi t \cos\left(\frac{\pi t}{1+t}\right) > (1+t) \sin\left(\frac{\pi t}{1+t}\right), 0 < t < \frac{1}{2}.$$

Ce qui achève la preuve. ■

**Lemme 3.3.2** Soit  $\Psi(t)$  la fonction noyau définie dans (3.13), alors on a

1.  $\Psi''(t) > 2, \forall t > 0.$
2.  $t\Psi''(t) + \Psi'(t) > 0, \forall t > 0.$
3.  $t\Psi''(t) - \Psi'(t) > 0, \forall t > 0.$
4.  $\Psi'''(t) < 0, \forall t > 0.$

**Preuve.**

1) En commençant d'abord par simplifier l'expression de  $\Psi''(t)$ , on a d'après (3.15)

$$\Psi''(t) = 2 + \frac{u'(t)^2 \sin^2(u(t)) - u''(t) \cos(u(t)) \sin(u(t)) + 2u'(t) \cos(u(t))}{\sin^3(u(t))},$$

Avec

$$u(t) = \frac{\pi t}{1+t}, u'(t) = \frac{\pi}{(1+t)^2}, u''(t) = -\frac{2\pi}{(1+t)^3},$$

Donc

$$\begin{aligned} \Psi''(t) &= 2 + \frac{\frac{\pi^2}{(1+t)^4} \sin^2(u(t)) - u''(t) \sin(u(t)) \cos(u(t)) + 2u'(t) \cos^2(u(t))}{\sin^3(u(t))} \\ &= 2 + \frac{\frac{\pi^2}{(1+t)^4} \sin^2(u(t)) + \frac{2\pi}{(1+t)^3} \sin(u(t)) \cos(u(t)) + 2\frac{\pi^2}{(1+t)^4} \cos^2(u(t))}{\sin^3(u(t))} \\ &= 2 + \frac{\pi^2}{(1+t)^4 \sin^3(u(t))} \left[ 1 + \frac{2(1+t)}{\pi} \sin(u(t)) \cos(u(t)) + \cos^2(u(t)) \right]. \end{aligned}$$

On distingue deux cas :

**Cas1** : pour  $0 < t \leq 1$ , On a

$$\sin(u(t)) > 0 \text{ et } \cos(u(t)) > 0,$$

Alors

$$\Psi''(t) > 2.$$

**Cas2** : pour  $t > 1$ , on utilise le Lemme (3.3.1), on trouve

$$(1+t) \sin(u(t)) < \pi \implies \frac{1}{(1+t)^3 \sin^3(u(t))} > \frac{1}{\pi^3},$$

Ce qui donne

$$\begin{aligned} \Psi''(t) &> 2 + \frac{\pi^2}{\pi^3(1+t)} \left[ 1 + \frac{2(1+t)}{\pi} \sin(u(t)) \cos(u(t)) + \cos^2(u(t)) \right] \\ &> 2 + \frac{1}{\pi(1+t)} \left[ 1 + \frac{2(1+t)}{\pi} \sin(u(t)) \cos(u(t)) + \cos^2(u(t)) \right] > 2 + \frac{1}{\pi(1+t)} [1 + \cos(u(t))]^2 > 2 \end{aligned}$$

2) Commencant d'abord par simplifier l'expression  $t\Psi''(t) + \Psi'(t)$

On a d'après (3.15) et (3.17)

$$\begin{aligned}
 t\Psi''(t) + \Psi'(t) &= \frac{t\pi^2}{(1+t)^4 \sin^3(u(t))} \left[ 1 + \frac{2(1+t)}{\pi} \sin(u(t)) \cos(u(t)) + \cos^2(u(t)) \right] + 4t - 2 - \frac{\pi}{(1+t)^2} \frac{\cos(u(t))}{\sin^2(u(t))}, \\
 &= \frac{1}{(1+t)^4 \sin^3(u(t))} [2\pi t + 2(1+t)\pi t \sin(u(t)) \cos(u(t)) + t\pi^2 \cos^2(u(t)) + \\
 &\quad 4t(1+t)^4 \sin^3(u(t)) - 2(1+t)^4 \sin^3(u(t)) - \pi(1+t)^2 \sin(u(t)) \cos(u(t))], \\
 &= \frac{1}{(1+t)^4 \sin^3(u(t))} [2(1+t)\pi t \sin(u(t)) \cos(u(t)) + t\pi^2 \sin^2(u(t))] + \\
 &+ \frac{1}{(1+t)^4 \sin^3(u(t))} [4t(1+t)^4 \sin^3(u(t)) - 2(1+t)^4 \sin^3(u(t)) - \pi(1+t)^2 \cos(u(t)) \cdot \sin(u(t)) + 2t\pi^2 \cos^2(u(t))], \\
 \text{d'où}
 \end{aligned}$$

$$t\Psi''(t) + \Psi'(t) = \frac{1}{(1+t)^4 \sin^3(u(t))} h(t).$$

avec

$$\begin{aligned}
 h(t) &= 2(1+t)\pi t \sin(u(t)) \cos(u(t)) + t\pi^2 \sin^2(u(t)) + 4t(1+t)^4 \sin^3(u(t)) - 2(1+t)^4 \sin^3(u(t)) + \\
 &\quad -\pi(1+t)^2 \cos(u(t)) \cdot \sin(u(t)) + 2t\pi^2 \cos^2(u(t)),
 \end{aligned}$$

On distingue deux cas :

**Cas1** : pour  $0 < t \leq \frac{1}{2}$ , en utilisant le Lemme (3.3.1), on trouve

$$h(t) > l(t),$$

où

$$\begin{aligned}
 l(t) &= \pi(1+t) \sin^3(u(t)) + 2t^2\pi^2 \cos^2(u(t)) + 2t\pi^2 \cos^2(u(t)) - \pi(1+t)^2 \sin(u(t)) \cos(u(t)) + \\
 &\quad + (4t-2)(1+t)^4 \sin^3(u(t)).
 \end{aligned}$$

Ce qui implique

$$\begin{aligned}
 h(t) &> (1+t) \sin^3(u(t)) \left[ \pi + (4t-2)(1+t)^3 \right] + 2t\pi^2(1+t) \cos^2(u(t)) - \pi(1+t)^2 \sin(u(t)) \cos(u(t)) \\
 &> (1+t) \sin^3(u(t)) (\pi + (4t-2)(1+t)^3) + (1+t)\pi \cos(u(t)) [2\pi t \cos(u(t)) - (1+t) \sin(u(t))] > 0.
 \end{aligned}$$

La dernière inégalité est obtenue à partir de

$$\sin(u(t)) > 0 \text{ pour } 0 < t \leq \frac{1}{2}, \quad (3.19)$$

du Lemme (3.3.1), on a

$$(1+t)\pi \cos(u(t)) [2\pi t \cos(u(t)) - (1+t) \sin(u(t))] > (1+t)\pi \cos(u(t)) [(1+t) \sin(u(t)) - (1+t) \sin(u(t))] = 0. \quad (3.20)$$

et

$$\pi + (4t-2)(1+t)^3 = \pi + t^2(4t^2 + 10t + 6) + (t-1)^2 - t^2 - 3.$$

Pour  $0 < t \leq \frac{1}{2}$ , on obtient

$$4t^2 + 10t + 6 > 0,$$

Donc

$$\pi + (4t - 2)(1 + t)^3 > \pi - 3 + (t - 1)^2 > 0. \quad (3.21)$$

D'après (3.19), (3.20) et (3.21), on trouve

$$h(t) > 0$$

et par suite

$$t\Psi''(t) + \Psi'(t) > 0, 0 < t \leq \frac{1}{2}.$$

**Cas 2 :** Pour  $\frac{1}{2} < t \leq 1$ , on utilise le Lemme (3.3.1), on obtient

$$h(t) = \pi^2 t + 2(1+t)\pi t \sin(u(t)) \cos(u(t)) + \pi t^2 \cos^2(u(t)) - (1+t)^2 \pi \sin(u(t)) \cos(u(t)) + (4t-2)(1+t)^4 \sin^3(u(t)).$$

Donc

$$\begin{aligned} h(t) &> (1+t)\pi \sin(u(t)) + \pi(1+t) \sin(u(t)) \cos(u(t))(2t - (1+t)) + \pi^2 t \cos^2(u(t)) + (4t-2)(1+t)^4 \sin^3(u(t)), \\ &> (1+t)\pi \sin(u(t))[1 + (t-1) \cos(u(t))] + (4t-2) \cdot (1+t)^4 \cdot \sin^3(u(t)) > 0. \end{aligned}$$

La dernière inégalité est obtenue à partir de

$$\sin(u(t)) > 0 \text{ et } 1 + (t-1) \cos(u(t)) > 0, \forall t \in ]\frac{1}{2}, 1].$$

et comme

$$\frac{1}{(1+t)^4 \sin^3(u(t))} > 0,$$

On trouve

$$t\Psi''(t) + \Psi'(t) > 0, \frac{1}{2} < t \leq 1.$$

3) En suivant les mêmes démarches de la deuxième propriété, alors on a

$$\begin{aligned} t\Psi''(t) - \Psi'(t) &= 2t + \frac{t\pi^2}{(1+t)^4 \sin^3(u(t))} \left[ 1 + \frac{2(1+t)}{\pi} \sin(u(t)) \cos(u(t)) + \cos^2(u(t)) \right] + \\ &\quad -2t + 2 + \frac{\pi \cos(u(t))}{(1+t)^2 \sin^2(u(t))} \\ &= \frac{1}{(1+t)^4 \sin^3(u(t))} \left( t\pi^2 \sin^2(u(t)) + 2\pi t(1+t) \sin(u(t)) \cos(u(t)) + 2t\pi^2 \cos^2(u(t)) \right) + \\ &\quad + \frac{1}{(1+t)^4 \sin^3(u(t))} \left( 2(1+t)^4 \sin^3(u(t)) + \pi(1+t)^2 \sin(u(t)) \cos(u(t)) \right) \end{aligned}$$

On distingue deux cas :

**Cas1** : pour  $0 < t \leq 1$ , On a

$$u(t) = \frac{\pi t}{(1+t)} > 0, \text{ et } u'(t) = \frac{\pi}{(1+t)^2} > 0,$$

Donc  $u$  est une fonction croissante et comme  $u(0) = 0$ , et  $u(1) = \frac{\pi}{2}$ , alors  $0 < u(t) \leq \frac{\pi}{2}$ . Avec  $0 < t \leq 1$

Donc

$$\sin(u(t)) > 0, \cos(u(t)) > 0.$$

D'où

$$t\Psi''(t) - \Psi'(t) > 0, 0 < t \leq 1.$$

**Cas 2** : Pour  $t > 1$ , on utilise le lemme (3.3.1), on obtient

$$t\Psi''(t) - \Psi'(t) > b(t),$$

Où

$$\begin{aligned} b(t) &= \frac{1}{(1+t)^4 \sin^3(u(t))} [t\pi^2 \sin^2(u(t)) + 2t\pi^2 \cos^2(u(t)) - 2t\pi^2 \cos^2(u(t)) - (1+t)^3 \sin^2(u(t)) + 2(1+t)^4 \sin^3(u(t))] \\ &= \frac{1}{(1+t)^4 \sin^3(u(t))} [t\pi^2 \sin^2(u(t)) + ((2+2t) \sin(u(t)) - 1)(1+t)^3 \sin^2(u(t))]. \end{aligned}$$

On sait que  $u$  est une fonction croissante et comme  $t > 1$ , alors  $u(t) > \frac{\pi}{2}$ , ce qui donne  $\sin(u(t)) > 1$ .

Et par suite

$$(2+2t) \sin(u(t)) - 1 > 3.$$

Ce qui implique

$$t\Psi''(t) - \Psi'(t) > \frac{1}{(1+t)^4 \sin^3(u(t))} [t\pi^2 \sin^2(u(t)) + 3(1+t)^3 \sin^2(u(t))] > 0.$$

4) D'après la relation (3.17), on a

$$\begin{aligned} \Psi'''(t) &= \frac{1}{\sin^4(u(t))} \left[ \frac{-6\pi^2}{(1+t)^5} \sin^3(u(t)) + \frac{-5\pi^3}{(1+t)^6} \sin^2(u(t)) \cos(u(t)) - \frac{6\pi}{(1+t)^4} \sin^2(u(t)) \cos(u(t)) \right] \\ &\quad + \frac{1}{\sin^4(u(t))} \left[ -\frac{12\pi^2}{(1+t)^5} \sin(u(t)) \cos^2(u(t)) - \frac{6\pi^3}{(1+t)^6} \cos^3(u(t)) \right] \\ &= \frac{-\pi}{(1+t)^6 \sin^4(u(t))} h(t), \end{aligned}$$

avec

$$\begin{aligned} h(t) &= 6\pi(1+t) \sin^3(u(t)) + 5\pi^2 \sin^2(u(t)) \cos(u(t)) + 6(1+t)^2 \sin^2(u(t)) \cos(u(t)) + \\ &\quad + 12\pi(1+t) \sin(u(t)) \cos^2(u(t)) + 6\pi^2 \cos^3(u(t)). \end{aligned}$$

On distingue deux cas :

**Cas 1 :** Pour  $0 < t \leq 1$ , on a

$$u(t) = \frac{\pi t}{1+t} > 0 \text{ et } u'(t) = \frac{\pi}{(1+t)^2} > 0.$$

Donc  $u$  est une fonction croissante et comme  $u(0) = 0$  et  $u(1) = \frac{\pi}{2}$ , alors  $0 < u(t) \leq \frac{\pi}{2}$ .

Ce qui implique

$$0 < \sin(u(t)) < 1 \text{ et } 0 < \cos(u(t)) < 1.$$

D'où  $h(t) > 0$ . Ce qui donne  $\Psi'''(t) < 0$ .

**Cas 2 :** Pour  $t > 1$ , on a

$$h(t) = 6\pi(1+t)\sin^3(u(t)) + 5\pi^2\sin^2(u(t))\cos(u(t)) + 6(1+t)^2\sin^2(u(t))\cos(u(t)) + 12\pi(1+t)\sin(u(t))\cos^2(u(t)) + 6\pi^2\cos^3(u(t))$$

$$= 6\pi\sin^2(u(t))\left[(1+t)\sin(u(t)) + \frac{5\pi}{6}\cos(u(t))\right] + 6(1+t)\sin(u(t))\cos(u(t))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right] + 6\pi\cos^2(u(t))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right],$$

$$> 6\pi\sin^2(u(t))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right] + 6(1+t)\sin(u(t))\cos(u(t))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right] + 6\pi\cos^2(u(t))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right].$$

(3.23)

Car on a

$$\cos(u(t)) < 0 \implies \frac{5\pi}{6}\cos(u(t)) > \pi\cos(u(t)).$$

On obtient

$$h(t) > 6\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right]\left[\pi\sin^2(u(t)) + (1+t)\sin(u(t))\cos(u(t)) + \pi\cos^2(u(t))\right] > 6\pi\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right]\left[1 + \frac{1}{\pi}(1+t)\sin(u(t))\cos(u(t))\right].$$

On utilise le Lemme (3.3.1), on obtient

$$h(t) > 6\pi(1 + \cos(u(t)))\left[(1+t)\sin(u(t)) + \pi\cos(u(t))\right] > 0.$$

Puisque on a

$$(1+t)\sin(u(t)) + \pi\cos(u(t)) > 0.$$

Donc  $\Psi'''(t) < 0$ . ■

**Lemme 3.3.3** Pour  $t \geq 1$ , on a

1.  $\cos(u(t)) \geq 1 - t$ .
2.  $\frac{1}{\sin(u(t))} \leq t^2 - 2t + 2$ .

**Preuve.** On a

$$u(t) = \frac{\pi t}{(1+t)} \text{ et } u'(t) = \frac{\pi}{(1+t)^2} > 0.$$

Donc  $u$  est une fonction croissante avec  $u(1) = \frac{\pi}{2}$ , alors

$$\frac{\pi}{2} \leq u(t) \leq \pi \Rightarrow \sin(u(t)) \leq 1.$$

1) On pose

$$f(t) = \cos(u(t)) + t - 1.$$

Donc

$$f'(t) = \frac{-\pi \sin(u(t)) + (1+t)^2}{(1+t)^2} \geq \frac{-\pi + 4}{(1+t)^2} > 0, t > 1.$$

D'où  $f$  est une fonction croissante et comme  $f(t) \geq f(1) = 0$ , alors  $f(t) \geq 0$ . Ce qui donne

$$\cos(u(t)) \geq 1 - t, t \geq 1.$$

2) On pose

$$g(t) = t^2 - 2t + 2 - \frac{1}{\sin(u(t))}, \forall t \geq 1.$$

Alors

$$g'(t) = 2t - 2 + \frac{\pi \cos(u(t))}{(1+t)^2 \sin^2(u(t))}, \forall t \geq 1.$$

D'après (1), on obtient

$$\begin{aligned} g'(t) &\geq 2(t-1) - \frac{\pi(t-1)}{(1+t)^2 \sin^2(u(t))} = \frac{2(t-1)(1+t)^2 \sin^2(u(t)) - \pi(t-1)}{(1+t)^2 \sin^2(u(t))} \\ &\geq (8-\pi) \frac{(t-1)}{(1+t)^2 \sin^2(u(t))} \geq 0. \end{aligned}$$

Donc  $g$  est une fonction croissante, ce qui donne

$$g(t) \geq g(1) = 0,$$

D'où

$$t^2 - 2t + 2 \geq \frac{1}{\sin(u(t))}.$$

Ce qui complète la preuve ■

Les Lemmes suivants sont très utile dans l'analyse des algorithmes de points intérieurs basés sur les fonctions noyaux.

**Lemme 3.3.4** Soit  $\Psi(t)$  une fonction deux fois différentiables pour  $t > 0$ , alors les propriétés suivantes sont équivalentes

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

**Lemme 3.3.5** Soit  $\Psi(t)$  la fonction noyau définie dans (3.13), alors on a

1.  $\Psi(t)$  est exponentiellement convexe pour  $t > 0$ , i.e.,

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

2.  $\Psi''(t)$  est strictement décroissante pour  $t > 0$ .

**Preuve.**

1) Montrer que  $\forall t > 0, \Psi(t)$  est exponentiellement convexe est équivalent à montrer que

$$\Psi'(t) + t\Psi''(t) \geq 0, \forall t > 0.$$

D'après le lemme (3.3.2), on a

$$\Psi'(t) + t\Psi''(t) \geq 0, t > 0.$$

Donc,  $\Psi(t)$  est exponentiellement convexe.

2) D'après le lemme (3.3.2), on a

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

et par suite  $\Psi''(t)$  est strictement décroissante. ■

**Lemme 3.3.6** Soit  $\Psi(t)$  vérifie les propriétés suivantes

1.  $\Psi''(t), t > 0$  est strictement décroissante.
2.  $t\Psi''(t) - \Psi'(t) > 0, t > 0$ .

Alors

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

**Lemme 3.3.7** On suppose que  $\Psi(t)$  est la fonction noyau définie dans (3.13), alors on a

1.  $\Psi(t) < \frac{\Psi''(1)(t-1)^2}{2}, t > 1$ .
2.  $\Psi(t) \leq t\Psi'(t), t \geq 1$ .

**Preuve.**

1) On utilise le développement de Taylor en  $t = 1$ , avec

$$\Psi'(1) = \Psi(1) = 0, \Psi'''(t) < 0, \forall t > 0.$$

On obtient

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

2) On utilise le développement de Taylor des deux fonctions  $\Psi(t)$  et  $\Psi'(t)$  en  $t = 1$ , on obtient

$$\Psi(t) = \Psi(1) + \frac{\Psi'(c_1)(t-1)}{1}, 1 < c_1 < t,$$

et

$$t\Psi'(t) = t\Psi'(1) + \frac{t\Psi''(c_2)(t-1)}{1}, 1 < c_2 < t.$$

Donc

$$\Psi(t) - t\Psi'(t) = \frac{\Psi'(c_1)(t-1)}{1} - \frac{t\Psi''(c_2)(t-1)}{1} = -(t-1)(t\Psi''(c_2) - \Psi'(c_1)) \leq 0.$$

D'après le Lemme (3.3.2).

Ce qui achève la preuve. ■

**Théorème 3.3.1** Soit  $\rho : [0, +\infty[ \rightarrow [1, +\infty[$  la fonction inverse de la fonction  $\Psi(t)$  pour  $t \geq 1$ , on a

$$\delta(v) \geq \frac{1}{2}\Psi'(\rho(\Phi(v))).$$

**Lemme 3.3.8** Pour  $\Phi(v) \geq 1$ , on a

$$\delta(v) \geq \frac{1}{4}\sqrt{\Phi(v)}.$$

**Preuve.** On utilise (3.12) et le Lemme (3.3.2), on obtient

$$S = \Psi(t) = \int_1^t \int_1^\varepsilon \Psi''(\tau) d\tau d\varepsilon \geq \int_1^t \int_1^\varepsilon 2d\tau d\varepsilon = (t-1)^2.$$

Ce qui implique

$$\sqrt{S} + 1 \geq t = \rho(S).$$

D'après le Théorème (3.3.1) et le Lemme (3.3.7), on a pour  $S = \Phi(v)$

$$\begin{aligned} \delta(v) &\geq \frac{1}{2} \Psi'(\rho(\Phi(v))), \\ &\geq \frac{\Psi(\rho(\Phi(v)))}{2\rho(\Phi(v))} = \frac{\Phi(v)}{2\rho(\Phi(v))}, \\ &\geq \frac{\Phi(v)}{2(1 + \sqrt{\Phi(v)})}. \end{aligned}$$

et comme  $\Phi(v) \geq 1$ , on obtient

$$2 + 2\sqrt{\Phi(v)} \geq 4\sqrt{\Phi(v)},$$

et par suite

$$\frac{\Phi(v)}{2(1 + \sqrt{\Phi(v)})} \geq \frac{\Phi(v)}{4\sqrt{\Phi(v)}} = \frac{1}{4} \sqrt{\Phi(v)},$$

D'où

$$\delta(v) \geq \frac{1}{4} \sqrt{\Phi(v)}.$$

■

### 3.3.1 La borne supérieure de la fonction de proximité après une $\mu$ -mise à jour

Au début de chaque itération externe, juste avant la  $\mu$ -mise à jour avec le facteur  $(1 - \theta)$ , on a  $\Phi(v) \leq \tau$ . Grâce à la mise à jour de  $\mu$ , le vecteur  $v = \sqrt{\frac{xs}{\mu}}$  est divisé par le facteur  $\sqrt{1 - \theta}$ , avec  $0 < \theta < 1$  ce qui conduit à une augmentation de la valeur de  $\Phi(v)$ . Alors, pendant les itérations intérieures au cours de l'algorithme,  $\Phi(v)$  diminue jusqu'à  $\Phi(v) \leq \tau$ .

Donc au cours de l'algorithme, les valeurs les plus grandes de  $\Phi(v)$  apparaissent juste après les mises à jour de  $\mu$ .

Dans le reste de cette partie nous effectuons une estimation de la  $\mu$ -mise à jour de  $\Phi(v)$ .

**Lemme 3.3.9** *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  $t_1 = t_2 = 1$ .*

**Théorème 3.3.2** *Soit  $\rho$  l'inverse de la fonction  $\Psi(t)$  pour  $t \geq 1$  et pour un vecteur  $v > 0$  et  $\beta \geq 1$ , on a*

$$\Phi(\beta v) \leq n\Psi\left(\beta\rho\left(\frac{\Phi(v)}{n}\right)\right).$$

**Preuve.**

**Cas1 :** pour  $\beta > 1$ , on prend le problème de maximisation

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

On utilise les conditions d'optimalité du premier ordre

$$\Phi(\beta v) + y(z - \Phi(v)) = 0, \quad (3.24)$$

Où  $y > 0$  est le multiplicateur de Lagrange.

La dérivation de (3.24) par rapport à  $v$ , donne

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

ce qui est équivalent à

$$\begin{aligned} \beta \sum_{i=1}^n \Psi'(\beta v_i) - y \sum_{i=1}^n \Psi'(v_i) &= 0 \iff \sum_{i=1}^n \beta \Psi'(\beta v_i) - \sum_{i=1}^n y \Psi'(v_i) = 0 \\ &\iff \beta \Psi'(\beta v_i) - y \Psi'(v_i) = 0, \forall i = \overline{1, n} \\ &\iff \beta \Psi'(\beta v_i) = y \Psi'(v_i), \forall i = \overline{1, n}. \end{aligned} \quad (3.25)$$

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

$$\Psi'(v_i) > 0.$$

Donc  $v_i \neq 1$  pour  $z_i$  donné et  $v_i$  vérifie les contraintes du problème (3.25). i.e.,  $\Phi(v_i) = z_i, \forall i$ , avec

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

Donc

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

D'après le Lemme (3.3.9), on a

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

On prend

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

Puisque nous avons un problème de maximisation, alors on a

$$\begin{cases} \beta \Psi'(\beta v_i) > 0, \\ y \Psi'(v_i) > 0, y > 0. \end{cases}$$

On définit la fonction

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

D'après l'équation (3.25), on a

$$g(v_i) = \frac{\beta}{y}, \forall i,$$

Donc, on résulte que  $g$  est une fonction constante.

Maintenant, on dérive  $g$  on obtient

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

D'après le Lemme (3.3.6) on trouve

$$g'(t) > 0.$$

Donc  $g$  est strictement croissante d'où les  $v_i$  sont égaux.

Posons  $v_i = t, \forall i$ . Alors

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

Comme  $\rho$  est la fonction inverse de la fonction  $\Psi$ , alors

$$t = \rho\left(\frac{z}{n}\right).$$

D'où

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

Donc

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

**Cas2 :** pour  $\beta = 1$ , on obtient

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

■

**Corollaire 3.3.1** Soient  $0 < \theta < 1$  et  $\beta = \frac{1}{\sqrt{1-\theta}}$ , si  $\Phi(v) \leq \tau$ , alors on a

$$\Phi\left(\frac{v}{\sqrt{1-\theta}}\right) = \Phi(v_+) \leq n\Psi\left(\frac{\rho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right).$$

**Preuve.** On a  $\rho : [0, +\infty[ \rightarrow [1, +\infty[$  est une fonction croissante donc

$$\rho\left(\frac{\Phi(v)}{n}\right) \geq 1$$

et comme  $0 \leq \theta < 1$ , il devient

$$\frac{1}{\sqrt{1-\theta}} \geq 1,$$

Ce qui implique

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

Par l'utilisation du Théorème (3.3.2) avec  $\beta = \frac{1}{\sqrt{1-\theta}}$ , on obtient

$$\Phi\left(\frac{v}{\sqrt{1-\theta}}\right) \leq n\Psi\left(\frac{\rho\left(\frac{\Phi(v)}{n}\right)}{\sqrt{1-\theta}}\right).$$

et comme  $\Phi(v) \leq \tau$ , on a

$$\frac{\Phi(v)}{n} \leq \frac{\tau}{n},$$

Alors

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

Ce qui donne

$$\Phi(v_+) = \Phi\left(\frac{v}{\sqrt{1-\theta}}\right) \leq n\Psi\left(\frac{\rho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right) := L(n, \theta, \tau).$$

Avec  $L(n, \theta, \tau)$  est la borne supérieure de la fonction  $\Phi(v_+)$  et elle désigne aussi la valeur de  $\Phi(v)$  après une  $\mu$ - mise à jour. ■

### 3.3.2 Analyse de la complexité de l'algorithme

Dans les prochains Lemmes nous analysons la complexité de notre algorithme prototype.

#### Calcul du pas de déplacement

Le choix du pas de déplacement  $\alpha$  est un autre cruciale problème dans l'analyse de l'algorithme. Il doit être pris de telle sorte que la proximité des itérées au courant -centre permet d'améliorer d'une quantité suffisante. Dans cette partie, on cherche ce pas de déplacement  $\alpha > 0$  de telle façon la stricte faisabilité des nouveaux itérés est garantie.

Si on prend le pas de déplacement  $\alpha$ , alors les nouveaux itérés sont donnés par

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

et

$$s^+ = s + \alpha \Delta s = \frac{s}{v}\left(v + \alpha v \frac{\Delta s}{s}\right) = \frac{s}{v}(v + \alpha ds).$$

Sachant que

$$dx = \frac{v \Delta x}{x} \text{ et } ds = \frac{v \Delta s}{s}.$$

On sait que

$$v_+ = \sqrt{\frac{x^+s^+}{\mu}},$$

Donc

$$v_+^2 = \frac{x^+s^+}{\mu} = \frac{\frac{x}{v}(v + \alpha dx) \frac{s}{v}(v + \alpha ds)}{\mu} = (v + \alpha dx)(v + \alpha ds).$$

Alors

$$v_+ = \sqrt{(v + \alpha dx)(v + \alpha ds)}.$$

Pour tout  $\alpha > 0$ , on pose

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

où  $f(\alpha)$  est appelée la différence de la proximité entre le nouvel et l'ancien itéré.

$$\Phi(v_+) = \Phi(\sqrt{(v + \alpha dx)(v + \alpha ds)}).$$

D'après le Lemme (3.3.4), on obtient

$$\Phi(v_+) \leq \frac{\Phi(v + \alpha dx) + \Phi(v + \alpha ds)}{2},$$

ce qui implique

$$f(\alpha) + \Phi(v) = \Phi(v_+) \leq \frac{\Phi(v + \alpha dx) + \Phi(v + \alpha ds)}{2},$$

d'où

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

Donc

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

Où

$$\begin{cases} f(0) = \Phi(v_+) - \Phi(v) = \Phi(v) - \Phi(v) = 0, \\ f_1(0) = \frac{\Phi(v) + \Phi(v)}{2} - \Phi(v) = 0. \end{cases}$$

Ce qui implique

$$f(0) = f_1(0) = 0.$$

On a

$$f_1(\alpha) = \frac{\Phi(v + \alpha dx) + \Phi(v + \alpha ds)}{2} - \Phi(v) = \frac{1}{2} \sum_{i=1}^n [\Psi(v_i + \alpha dx_i) + \Psi(v_i + \alpha ds_i)] - \sum_{i=1}^n \Psi(v_i),$$

D'où

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

où  $dx_i$  et  $ds_i$  sont la  $i^{\text{ème}}$  composante des vecteurs  $dx$  et  $ds$  respectivement, et comme

$$dx + ds = -\nabla\Phi(v), \delta(v) = \frac{1}{2} \|\nabla\Phi(v)\|,$$

On obtient

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

**Lemme 3.3.10** Pour  $\delta := \delta(v)$  et  $v_{\min} := \min_i v_i$ , on a

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

**Lemme 3.3.11** Si le pas de déplacement vérifie l'inégalité suivante

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

alors

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

**Preuve.** On a, d'après le lemme (3.3.10)

$$f_1'(\alpha) = f_1'(0) + \int_0^\alpha f_1''(\varepsilon) d\varepsilon \leq -2\delta^2 + 2\delta^2 \int_0^\alpha \Psi''(v_{\min} - 2\varepsilon\delta) d\varepsilon.$$

On utilise le changement de variable suivant

$$y = v_{\min} - 2\varepsilon\delta,$$

ce qui implique

$$\varepsilon = \frac{y - v_{\min}}{2\delta},$$

alors

$$d\varepsilon = \frac{dy}{2\delta} \implies dy = 2\delta d\varepsilon.$$

On a

$$0 < \varepsilon < \alpha \implies v_{\min} < y < v_{\min} - 2\alpha\delta.$$

D'où

$$\begin{aligned} f_1'(\alpha) &\leq -2\delta^2 + \delta \int_{v_{\min}}^{v_{\min} - 2\alpha\delta} \Psi''(y) dy = -2\delta^2 - \delta \int_{v_{\min} - 2\alpha\delta}^{v_{\min}} \Psi''(y) dy, \\ &\leq -2\delta^2 - \delta(\Psi'(v_{\min}) - \Psi'(v_{\min} - 2\alpha\delta)) = -2\delta^2 + \delta(\Psi'(v_{\min} - 2\alpha\delta) - \Psi'(v_{\min})). \end{aligned}$$

Ce qui implique

$$f_1'(\alpha) \leq -2\delta^2 + 2\delta^2 = 0.$$

■

**Lemme 3.3.12** Soit  $\rho : [0, +\infty[ \rightarrow ]0, 1]$  l'inverse de la fonction  $-\frac{1}{2}\Psi'(t)$  pour tout  $t \in ]0, 1]$ , alors la valeur maximale du pas de déplacement  $\alpha$  qui vérifie l'inégalité

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

est donnée par

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

**Lemme 3.3.13** Soient  $\rho$  et  $\bar{\alpha}$  qui sont définis dans le Lemme (3.3.12), alors

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

**Preuve.** Par définition de  $\rho$ , on a

$$\rho(\delta) = \left(-\frac{\Psi'(\delta)}{2}\right)^{-1}.$$

Donc

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

Ce qui donne

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

On dérive par rapport à  $\delta$ , on obtient

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

d'où

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

Alors  $\rho$  est une fonction décroissante. D'après le Lemme (3.3.12), on a

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

Comme  $\rho$  est décroissante, alors

$$\rho(2\delta) = \min_{[\delta, 2\delta]} \rho(\delta).$$

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

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

D'où

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

■

Pour trouver une borne supérieure de  $f(\alpha)$  nous avons besoin d'un pas de déplacement par défaut  $\tilde{\alpha}$  qui est la borne inférieure de  $\bar{\alpha}$ .

**Lemme 3.3.14** Soit  $\rho : [0, +\infty[ \rightarrow ]0, 1]$  la fonction inverse de la restriction  $-\frac{1}{2}\Psi'(t)$  sur l'intervalle  $]0, 1]$  et  $1 \leq \tau \leq \Phi(v)$ , donc on a

$$\frac{1}{\Psi''(\rho(2\delta))} \geq \frac{1}{(16 + 24\sqrt{6}\pi^2)\delta^{\frac{3}{2}}} : = \tilde{\alpha}.$$

**Lemme 3.3.15** On suppose que  $g$  est une fonction convexe et deux fois différentiable avec

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

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 3.3.16** Si le pas de déplacement  $\alpha$  vérifie la condition  $\alpha \leq \bar{\alpha}$ , alors

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

**Preuve.** Soit  $g$  une fonction définie par

$$g(\alpha) = -2\alpha\delta^2 + \alpha\delta\Psi'(v_{\min}) - \frac{1}{2}\Psi(v_{\min}) + \frac{1}{2}\Psi(v_{\min} - 2\alpha\delta).$$

Alors

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

Par dérivation, on trouve

$$g'(\alpha) = -2\delta^2 + \delta\Psi'(v_{\min}) - \delta\Psi'(v_{\min} - 2\alpha\delta).$$

Alors

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

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

$$g''(\alpha) = 2\delta^2\Psi''(v_{\min} - 2\alpha\delta).$$

D'où

$$f''(\alpha) \leq g''(\alpha)$$

d'après le Lemme (3.3.10).

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

$$g'(\alpha) = g'(0) + \int_0^\alpha g''(\varepsilon)d\varepsilon = -2\delta^2 + 2\delta^2 \int_0^\alpha \Psi''(v_{\min} - 2\varepsilon\delta)d\varepsilon = -2\delta^2 - \delta \int_0^\alpha \Psi''(v_{\min} - 2\varepsilon\delta)d(v_{\min} - 2\varepsilon\delta)$$

$$g'(\alpha) = -2\delta^2 + \delta(-\Psi'(v_{\min} - 2\alpha\delta) + \Psi'(v_{\min})).$$

On utilise le Lemme (3.3.11), on obtient

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

Avec  $g''(\alpha)$  est croissante, donc d'après le Lemme (3.3.15), on a

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

D'où

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

■

**Théorème 3.3.3** Si  $\alpha = \tilde{\alpha}$  le pas de déplacement par défaut qui est défini par  $\tilde{\alpha} = \frac{1}{(16+24\sqrt{6}\pi^2)\delta^{\frac{3}{2}}}$ , alors

$$f(\tilde{\alpha}) \leq -\frac{\Phi^{\frac{1}{4}}}{32 + 48\sqrt{6}\pi^2}.$$

**Proposition 3.3.1** On pose  $\alpha = \tilde{\alpha}$  dans le Lemme (3.3.16), on obtient

$$f(\tilde{\alpha}) \leq -\tilde{\alpha}\delta^2 = -\frac{\delta^2}{(16 + 24\sqrt{6}\pi^2)\delta^{\frac{3}{2}}} = -\frac{\delta^{\frac{1}{2}}}{16 + 24\sqrt{6}\pi^2}.$$

En utilisant le Lemme (3.3.8), on trouve

$$f(\tilde{\alpha}) \leq -\frac{\Phi^{\frac{1}{4}}}{2(16 + 24\sqrt{6}\pi^2)} = -\frac{\Phi^{\frac{1}{4}}}{32 + 48\sqrt{6}\pi^2}.$$

### 3.3.3 Nombre total d'itérations

Nous avons besoin de compter combien d'itération interne est nécessaire pour retourner à la situation  $\Phi(v) \leq \tau$  après une  $\mu$ -mise à jour. On définit la valeur de  $\Phi(v)$  après la première mise à jour de  $\mu$  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 interne dans une itération externe. Par la diminution de  $f(\tilde{\alpha})$  pour  $k = \overline{1, K-1}$ , on obtient

$$\Phi_{k+1}(v) \leq \Phi_k(v) - \frac{\Phi(v)^{\frac{1}{4}}}{32 + 48\sqrt{6}\pi^2}.$$

**Lemme 3.3.17** Soit  $t_0, t_1, \dots, t_k$  une suite des nombres positives, avec

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

alors

$$K \leq \frac{t_0^\varepsilon}{\beta \varepsilon}.$$

Si on prend  $t_k = \Phi_k(v)$ ,  $\varepsilon = \frac{3}{4}t$  et  $\beta = \frac{1}{32+48\sqrt{6\pi^2}}$ ,

nous pouvons obtenir le théorème suivant

**Théorème 3.3.4** Soit  $k$  le nombre total d'itérations interne dans une itération externe, alors on a

$$k \leq \frac{4}{3}(32 + 48\sqrt{6\pi^2})\Phi_0^{\frac{3}{4}}.$$

Où  $\Phi_0$  est la valeur de  $\Phi(v)$  après la première  $\mu$ -mise à jour dans une itération externe.

**Preuve.** D'après le Lemme (3.3.3), il est claire que

$$\Psi(t) \leq 2(t-1)^2, \forall t \geq 1.$$

En appliquant le Corollaire (3.3.1) et la Preuve du Lemme (3.3.8), on obtient

$$\begin{aligned} \Phi_0 &\leq n\Psi\left(\frac{\rho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right) \leq n\Psi\left(\frac{1+\sqrt{\frac{\tau}{n}}}{\sqrt{1-\theta}}\right) \\ &\leq 2n\left(\frac{1+\sqrt{\frac{\tau}{n}}-\sqrt{1-\theta}}{\sqrt{1-\theta}}\right)^2 \leq \frac{2n}{1-\theta}\left(\theta + \sqrt{\frac{\tau}{n}}\right)^2. \end{aligned}$$

Où la dernière inégalité est obtenue à partir de

$$1 - \sqrt{1-\theta} \leq \theta \text{ pour } 0 < \theta \leq 1.$$

■

**Théorème 3.3.5** Si  $\tau \geq 1$  et  $n\mu < \varepsilon$ , alors le nombre total d'itérations pour trouver une solution optimale primale-duale ne dépasse pas la valeur

$$4\left(\frac{32 + 48\sqrt{6\pi^2}}{3\theta}\right)\left(\frac{2n}{1-\theta}\left(\theta + \sqrt{\frac{\tau}{n}}\right)^2\right)^{\frac{3}{4}} \log\left(\frac{n}{\varepsilon}\right).$$

**Preuve.** On commence par déterminer le nombre d'itérations externes  $K$ . On a dans l'algorithme  $n\mu \geq \varepsilon$ . Avec

$$\mu_K = (1-\theta)^K \mu_0, \mu_0 = 1$$

Donc

$$n(1-\theta)^K \mu_0 \geq \varepsilon$$

Ce qui implique

$$(1-\theta)^K \geq \frac{\varepsilon}{n}$$

Alors

$$K \log(1 - \theta) \geq \log\left(\frac{\varepsilon}{n}\right)$$

Puisque

$$-\log(1 - \theta) \geq \theta, 0 < \theta < 1$$

On obtient

$$-\log\left(\frac{\varepsilon}{n}\right) \geq -K \log(1 - \theta) \geq K\theta.$$

Alors

$$K \leq -\frac{1}{\theta} \log\left(\frac{\varepsilon}{n}\right) = \frac{1}{\theta} \log\left(\frac{n}{\varepsilon}\right).$$

Et par suite le nombre d'itérations extérieures est borné par  $\frac{1}{\theta} \log\left(\frac{n}{\varepsilon}\right)$ . Donc le nombre total d'itérations est obtenu par la multiplication de nombres d'itérations interne et le nombre d'itérations externe. Ce qui termine la preuve. ■

**Remarque 3.3.1** 1. Si on prend  $\tau = O(n)$  et  $\theta = \Theta(1)$ , alors le nombre d'itérations pour les algorithmes à grand-pas est d'ordre  $O\left(n^{\frac{3}{4}} \log\left(\frac{n}{\varepsilon}\right)\right)$ .

2. Si  $\tau = O(1)$  et  $\theta = \Theta(n)$ , alors la complexité pour les algorithmes à petit-pas est  $O\left(\sqrt{n} \log\left(\frac{n}{\varepsilon}\right)\right)$ .

# Conclusion Générale

D'après les travaux des chercheurs jusqu'à l'année 2015, La meilleure complexité atteinte par les algorithmes basés sur les fonctions noyaux est bornée par  $O(\sqrt{n}(\log(n)) \log(\frac{n}{\epsilon}))$  ([10]) itérations pour les méthodes de points intérieurs de la programmation linéaire à grand-pas, par contre **B.KHEIRFAM** et **M.MOSLEMI** ont réussi à améliorer cette complexité qui devient  $O(n^{\frac{3}{4}} \log(\frac{n}{\epsilon}))$  par l'utilisation d'une nouvelle fonction noyau de type trigonométrique.

## **Perspectives :**

Comme perspectives, on pense à

1. Programmer l'algorithme de notre approche.
2. Faire une étude comparative avec d'autres fonctions noyaux.

# Bibliographie

- [1] N. Anane, *méthodes de point intérieurs pour la programmation linéaire basées sur les fonctions noyaux*. Mémoire de magister, université de farhat abbas-sétif, (2012).
- [2] Y.Q. Bai, M. Elghami, C. Roos, *A comparative study of karnel function for primal-dual interior point algorithm in linear optimisation*, SIAM Journal on Optimization. 15, 101 – 128 (2005).
- [3] M. Bouafia, D. Bentarki, A. Yassine, *An efficient primal dual interior point méthode for linear programing problemes based on a new karnel function with trigonometric barrier term*. Journal of Optimization Theory and Applications. 170, 528 – 545(2016).
- [4] M. Elghami, *New primal dual interior point méthodes based on karnel functions, certificat d'études approfondies mathématiques (C.E.A)*. Université mohammed V, Rabat. (2005).
- [5] B. Kheirfam, M. Moslemi, *A polynomial-time algorithm for linear optimization based on a new karnel function with trigonometric barrier term*. Yugoslav Journal of Operations Research. 25, 233 – 250(2014).
- [6] M. Minoux, *Programmation mathématique : théorie et algorithmes*. Tome 1 Dunod. Paris (1983).
- [7] H. Mellit, I. Merghit, *Résolution du problème SDP par une class de IPMS*, mémoire de master, université de Jijel, Algérie (2016).
- [8] L. Menniche, *Etude théorique d'une classe de méthodes de points intérieurs pour la programmation linéaire*. Thèse de Doctorat, université de Sétif (2017).
- [9] I. Touil, *Etude comparative des performances d'une méthode de point intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire*. Mémoire de magister, université de Ferhat Abbas, Sétif, Algérie. (2005).
- [10] I. Touil, *Etude théorique et numéruque des méthodes de points intérieurs de type trajectoire centrale pour la programmation semi-définie linéaire*. Thèse de Doctorat, université de Sétif (2017).
- [11] L. Xin, Z. Mingwang, *Interior-point algorithm for linear optimization based on a new trigonometric kernel function*. Operations Research Letters. 43, 471 – 475(2015).