



Ministère de l'Enseignement Supérieur et de
la Recherche Scientifique
Université Mohamed El-Bachir El-Ibrahimi
Bordj Bou Arréridj
Faculté des mathématiques et de
l'informatique
Département de Recherche Opérationnelle



N° ordre.....

Mémoire de Master

Mémoire présenté en vue de l'obtention du diplôme de Master en :

Domaine : Mathématiques et Informatique.

Filière : Mathématiques.

Spécialité : Méthodes et outils pour la recherche opérationnelle.

*Implémentation numérique d'un algorithme de
trajectoire central avec poids basé sur une nouvelle
direction de recherche pour la programmation
linéaire.*

Présenté par :

- Hammoudi khadidja
- Belfar Hadjer

Sous la direction de :
Mme. GUERRA LOUBNA

Soutenu publiquement 15/Juin/2023 devant le jury composé de :

M.	Belkacem Nazihddine	Maitre Assistant A	Université de BBA	Président
M.	Ghebouli Messaoud	Maître de Conférence B	Université de BBA	Examineur
Mme.	Guerra Loubna	Maître de Conférence B	Université de BBA	Encadreur

Année Universitaire

2022/2023



Remerciements

*A*vant tout on remercie " **Allah** " tout puissant pour la volonté, la patience que nous a données d'accomplir ce travail.

*U*n grand remerciement à notre encadreur " **Dr. GUERRA Loubna** "; maître de conférence classe B; à l'université Mohamed El-Bachir-El-Ibrahimi, Bordj Bou Arréridj; pour avoir accepté de diriger ce mémoire, et pour les conseils fructueux et sa disposition, l'encouragement et l'orientations qu'elle nous a attribuées.

*N*ous remercions les membres du jury pour leur présence, pour leur lecture attentive de ce mémoire, ainsi que pour les remarques qu'ils m'adresseront lors de cette soutenance afin d'améliorer mon travail.

*N*ous remercions tous les enseignants de la faculté des mathématiques et de l'informatique de l'université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj.

*M*erci à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce mémoire. Votre soutien a été précieux et a joué un rôle majeur dans la réussite de ce projet.

Dédicace

Je dédie ce travail :

A "ma très chère mère" , qui me donne toujours l'espoir de vivre et qui n'a jamais cessé de prier pour moi.

A "ma très chère père " , pour ses encouragements, son soutien, et son sacrifice afin que rien n'entrave le déroulement de mes études.

A mes frères et mes sœurs "Lotfi et Ayoub mon jumeau Sara Ahlem Khouloud Raïfa Malak".

Vous avez toujours été là pour moi, me soutenant et m'encourageant à chaque étape de mon parcours académique. Votre amour, votre soutien et votre présence ont été des piliers essentiels de ma réussite.

A mon fiancé "Amine"

dédicace sincère pour toi. Merci d'être ma lumière et ma force. Ton amour inconditionnel m'a porté jusqu'à cette réussite. Je t'aime infiniment.

A ma chère amie "Marwa"

Tu es bien plus qu'une simple amie, tu es ma famille choisie. Je serai toujours là pour toi et je t'aime énormément.

A ma chère binome Khadidja pour monnre longue amitié.

A tout mes amies " Noussa Khaled Amar Baker Ayoub Fadia Wael Amina Wiam Bouchra Haizi Fatima Assma Yssine Abdkrim Alla Sara .." et tout mes enseignants à la faculté MI et en général, tous ceux que j'aime et respecte.

Merci!

Belfar hadjer

Dédicace

Avec l'expression de ma reconnaissance, je dédie ce modeste travail :

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon père "OMAR".

A la femme qui a souffrit sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse : ma mère "MALIKA".

A mes chères SOEURS et leurs ENFANTS" qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études. Que Dieu les protège et leurs offre la chance et le bonheur.

A mon seul frère "HICHEM" qui m'a toujours soutenu et encouragé durant ces années d'études.

A ma belle collègue "NADJAT" Pour votre appui, vos encouragements pour les précieux conseils puisse dieu vous donne tous que vous voulez.

Sans oublier m'adorable cousine "NASSIMA" puisse dieu vous donne la santé, le bonheur, le courage, et surtout la réussite.

Merci!

Hammoudi khadidja

Table des matières

Notations utilisées	2
Terminologie	3
Introduction générale	4
Introduction	6
1 Notions de Base pour la programmation linéaire	6
1.1 Ensembles et fonctions	6
1.1.1 Ensemble et fonction borné	6
1.1.2 Ensemble et fonction convexe	7
1.1.3 Ensemble et fonction affine	10
1.1.4 Matrices et vecteurs	11
1.1.5 Matrices	12
1.1.6 Convexité et dérivée	13
1.2 Programmation mathématique	13
1.2.1 Définitions	13
1.2.2 Classification d'un programme mathématique	14
1.2.3 Qualification des contraintes	15
1.2.4 Existence et unicité	15
1.2.5 Conditions d'optimalités	15
1.3 Méthode de Newton	15
1.3.1 Interprétation graphique	16
2 Méthode de Trajectoire Centrale avec poids pour PL	19
2.1 Position de problème	19
2.1.1 Dualité faible	20
2.1.2 Dualité forte	21
2.2 Modélisation de quelques problèmes réels sous forme d'un PL	21
2.3 Méthode de trajectoire centrale	23
2.3.1 La mesure de proximité	26
2.3.2 La description de l'algorithme	26
2.3.3 Algorithme TC classique	26
2.4 Méthode de TC avec Poids	27
2.4.1 Nouvelle direction de recherche	28
2.4.2 Algorithme	29
2.4.3 Convergence de l'algorithme	30
2.4.4 Analyse de la complexité	34

3	Implémentation numérique	36
3.1	Tests numériques	37
	Bibliographie	46

Notations utilisées

\mathbb{R}^n	:	L'espace des vecteurs réels de dimension n ,
\mathbb{R}_+^n	:	L'orthant positif de \mathbb{R}^n ,
$\mathbb{R}^{m \times n}$:	L'espace vectoriel des matrices réelles de taille $(m \times n)$,
$\ \cdot\ $:	Norme euclidienne sur \mathbb{R}^n ,
$\text{rg}(A)$:	Le rang de la matrice $A \in \mathbb{R}^{m \times n}$,
A^t	:	Le transposé d'une matrice $A \in \mathbb{R}^{m \times n}$,
a_{ij}	:	Élément de la matrice $A \in \mathbb{R}^{m \times n}$,
x^t	:	Transposé d'un vecteur $x \in \mathbb{R}^n$,
xz	=	$x_i z_i, \forall i$,
$\frac{x}{z}$	=	$\frac{x_i}{z_i}, \forall i; z_i \neq 0$,
x^2	=	$x_i^2, \forall i$,
x^{-1}	=	$\frac{1}{x_i} (x_i \neq 0, \forall i)$,
$x \geq 0 (x > 0)$	=	$x_i \geq 0 (x_i > 0), \forall i$,
$x \leq 0 (x < 0)$	=	$x_i \leq 0 (x_i < 0), \forall i$,
\sqrt{x}	=	$\sqrt{x_i} (x_i > 0, \forall i)$,
e	:	Un vecteur de \mathbb{R}^n tel que $e_i = 1, \forall i = 1, \dots, n$,
$\frac{1}{2}e$	=	$(\frac{1}{2}, \dots, \frac{1}{2})$,
$\Delta x, \Delta z$:	les directions de Newton,
$0_{m \times n}$:	La matrice zéro $(m \times n)$,
I_m	:	La matrice identité de taille $(m \times m)$,
$\min(x)$:	$\min(x_i), i = 1, \dots, n$,
$\text{diag}(x)$	=	X la matrice diagonale avec $X_{ii} = x_i$,
$f(x) = O(g(x))$	\Leftrightarrow	$\exists k > 0 : f(x) \leq k g(x)$, pour tout $x > 0$.

Terminologie

PM	: Programmation mathématique,
PL	: Programmation linéaire,
(P)	: Le problème linéaire sous forme primal,
(D)	: Le problème dual de (P),
K.K.T	: Karush-Kuhn-Tucker,
TC	: Trajectoire centrale,
CPI	: Condition de points intérieurs.

Introduction générale

La programmation mathématique est une branche d'optimisation, s'occupe de la minimisation sous contraintes d'une fonction à plusieurs variables, schéma très général s'appliquant à de nombreuses situations pratiques dans beaucoup de domaines (minimisation de coûts, de durées, etc.)

Dans le cas d'une fonction est de contraintes linéaires on parle de la programmation linéaire (*PL*), cette dernière a été développée en **1947** et utilisée par **Géorge Bernard Danzing**. La programmation linéaire est un cadre mathématique général permettant de modéliser et de résoudre certains problèmes d'optimisation, qu'est consisté à trouver le maximum ou le minimum d'une forme linéaire dite fonction objectif en satisfaisant certaines équations et inégalités dites contraintes. Parmi les méthodes de résolution d'un problème linéaire on cite :

- **Méthodes graphique.**
- **Méthodes du simplexe.**
- **Méthodes de points intérieurs .**

L'une des méthodes pour résoudre les problèmes linéaires en nombre réels est la méthode classique du simplexe, qui est la plus utilisée depuis longtemps. Cette méthode a été découverte en **1947** par **George Danzing**. La méthode du simplexe est une procédure itérative permettant d'effectuer une exploration dirigée de l'ensemble des solutions réalisables de base. L'application de cette méthode nécessite la connaissance d'une solution de base réalisable de départ, ensuite on utilise la méthode des tableaux pour obtenir la solution optimale. Il est montré par [23] que cette méthode a une complexité exponentielle. Pour cela, on utilise les méthodes de points intérieurs qui ont été initialisées pour la première fois par **Karmakar** en **1984**. Depuis cette date, de nombreux algorithmes ont été développés pour la programmation linéaire [11, 19]. On distingue quatre classes fondamentales de méthodes de points intérieurs à savoir :

- **Méthodes affines.**
- **Méthodes projectives.**
- **Méthodes de réduction de potentiel.**
- **Méthodes de la trajectoire centrale.**

La méthode de trajectoire centrale a été introduite à la même époque que les méthodes de réduction de potentiel et pleinement développées au début des années **90**. Elles possèdent de bonnes propriétés théoriques : une complexité polynomiale et une convergence linéaire [19]. L'extension pour d'autres problèmes d'optimisation, pour plus de détail voir [1, 23]. Ils utilisent les directions de Newton classiques pour obtenir le nouveau itéré. Roos et al en 1997 [19] ont montré que cet algorithme a une complexité polynomiale. Dont le but est de généraliser la méthode de trajectoire centrale classique qui est déjà présentée par Roos 1997 [19] et d'avoir nouvelles directions de recherche, Dravay en **2002** [6] a été proposé une nouvelle transformation algébrique $\psi(xz) = \psi(w^2)$ où $\psi(t) = \sqrt{t}$ au lieu d'utiliser l'équation de centralité $xz = \mu e$, avec $xz = w^2$ Dravay a montré théoriquement que la complexité de l'algorithme proposé est donnée par

$$O \left[\frac{5}{2} \sqrt{\sigma_c(w^0)} n \log \frac{(x^0)^t z^0}{\epsilon} \right],$$

où $\sigma_c(w^0) = \frac{\max((w^0)^2)}{\min((w^0)^2)}$, mais il a montré théoriquement que la meilleure borne est obtenue si en suit la TC classique c'est à dire si $\sigma_c(w^0) = 1$ avec une complexité

$$O \left[\frac{5}{2} \sqrt{n} \log \frac{(x^0)^t z^0}{\epsilon} \right].$$

On s'intéresse à implémenter numériquement l'algorithme qui présenté par Darvay selon le choix de $\sigma_c(w^0)$ et voir son l'influence sur le comportement numérique.

Ce mémoire est composé de trois chapitre :

- 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 programmation mathématique, les principaux résultats d'existence, les conditions d'optimalité et la programmation linéaire.
- Dans **le deuxième chapitre**, on va étudier un algorithme de trajectoire central de type primal-dual classique qui présenté par **Roos** pour résoudre la programmation linéaire, puis on étudié un algorithme de trajectoire central de type Primal-dual avec poids qui est présenté par Darvay [6]. Dont la complexité de cet algorithme est polynomial.
- Dans **le troisième chapitre**, on présente les tests numériques appliqués sur quelques problèmes linéaire, avec comparaison entre différents valeurs de w^0 pour prouver les résultats théoriques. On terminer ce mémoire par une conclusion générale.

1

Notions de Base pour la programmation linéaire

Dans ce chapitre, on va présenter quelques notions de base essentielles d'analyse convexe, d'algèbre linéaire et la programmation mathématique, qui permettent de résoudre de nombreux problèmes. Et on va introduire également des concepts qui jouent un rôle très important : tels que la convexité des fonctions et des ensembles, ainsi que la méthode de Newton, ... etc."

1.1 Ensembles et fonctions

1.1.1 Ensemble et fonction borné

Définition 1.1. (*Ensemble borné*)

Soit $S \subset \mathbb{R}^n$ un ensemble

- S est dit **majoré** s'il existe un élément $M \in \mathbb{R}$ tel que

$$x \in S, \quad x \leq M.$$

- S est dit **minoré** s'il existe un élément $m \in \mathbb{R}$ tel que

$$x \in S, \quad x \geq m.$$

- S est **bornée** si S est à la fois **majorée** et **minorée**.

Définition 1.2. (*Fonction borné*)

Soit $f : S \rightarrow \mathbb{R}$ une fonction

- f est **majorée** sur S s'il existe $M \in \mathbb{R}$ tel que $f(x) \leq M$ pour tout $x \in S$. On dit alors que M est un **majorant** de f .
- f est **minorée** sur S s'il existe $m \in \mathbb{R}$ tel que $f(x) \geq m$ pour tout $x \in S$. On dit alors que m est un **minorant** de f .
- f est **bornée** sur S si f est à la fois **majorée** et **minorée** sur S .

Définition 1.3. (*sous-ensemble compact*)

Un sous-ensemble $S \subset \mathbb{R}^n$ est dit compact si et seulement si il est fermé et borné (c'est à-dire contenu dans une boule de rayon $M < +\infty$).

1.1.2 Ensemble et fonction convexe

La notion de convexité est un outil mathématique d'importance capitale pour l'étude théorique et numérique des problèmes d'optimisation. A ce propos, on présente dans ce paragraphe quelques notions de base d'usage courant.

Définition 1.4. :

Un ensemble $S \subset \mathbb{R}^n$ est dit convexe si et seulement si :

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

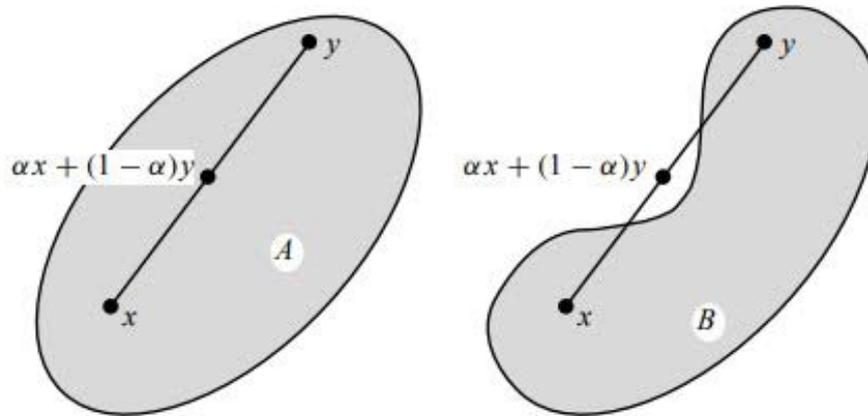


FIGURE 1.1: Ensemble convexe et non convexe

Définition 1.5. On peut dire que S est convexe si et seulement si pour deux points quelconques x et y pris dans S , le segment $[x,y]$ est tout entier contenu dans S .

Définition 1.6. (*combinaison convexe*)

Etant donné p points de $\mathbb{R}^n(x_1, x_2, \dots, x_p)$ on dit que $x \in \mathbb{R}^n$ est une combinaison convexe de ces p points s'il existe des coefficients $\mu_1, \mu_2, \dots, \mu_p$ ($\mu_i \geq 0, \forall i = 1, 2, \dots, p$) tels que :

$$x = \sum_{i=1}^p \mu_i x_i \quad \text{et} \quad \sum_{i=1}^p \mu_i = 1.$$

Proposition 1.1. .

Les opérations algébriques suivantes conservent la convexité :

1. L'intersection quelconque : Pour tout A, B convexes alors $A \cap B$ est convexe.
2. Le produit cartésien : Pour tout A, B convexes alors $A \times B$ convexe.
3. les transformations affines : Pour tout C convexe alors $\alpha C + \beta$ est convexe $\forall \alpha \in \mathbb{R}^*$ et $\beta \in \mathbb{R}$.

4. Les combinaisons linéaires : Pour tout C_i convexe alors $\sum_{i=1}^m \alpha_i C_i$, ($\forall \alpha_i \in \mathbb{R}$ et $m \in \mathbb{N}$).
5. La translation $C + a$ avec $a \in \mathbb{R}^n$.

Définition 1.7. Soit $S \subseteq \mathbb{R}^n$ un ensemble convexe On dit qu'une fonction $f : S \rightarrow \mathbb{R}$, est convexe, si elle vérifié :

$$\forall (x, y) \in S, \quad \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (1.1)$$

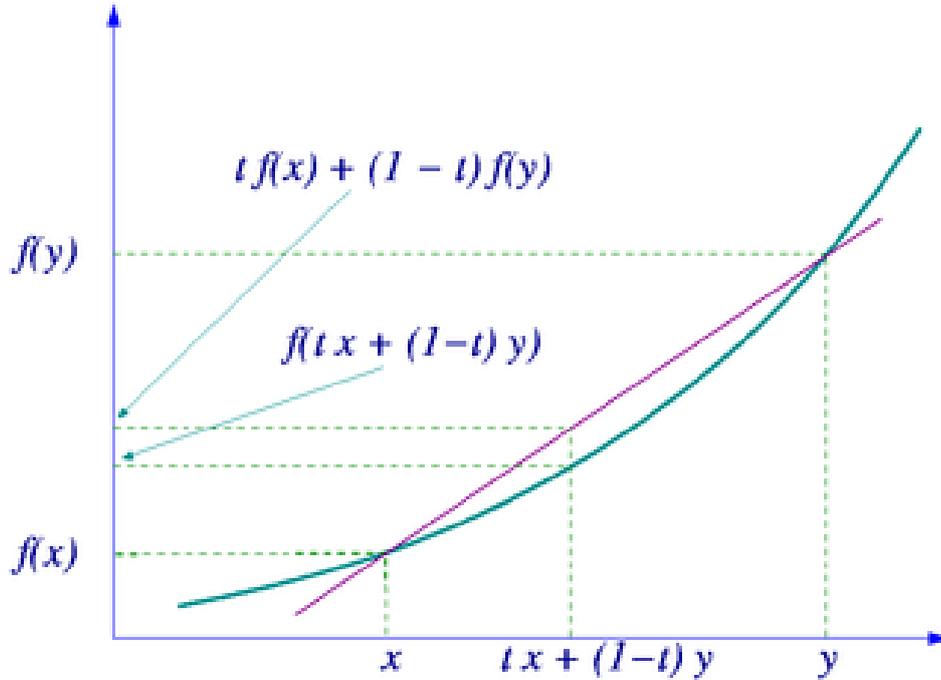


FIGURE 1.2: Fonction convexe

Définition 1.8. Soit $S \subseteq \mathbb{R}^n$ un ensemble convexe, on dit qu'une fonction $f : S \rightarrow \mathbb{R}$ définie sur S convexe, est concave, si elle vérifié :

$$\forall (x, y) \in S, \quad \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y). \quad (1.2)$$

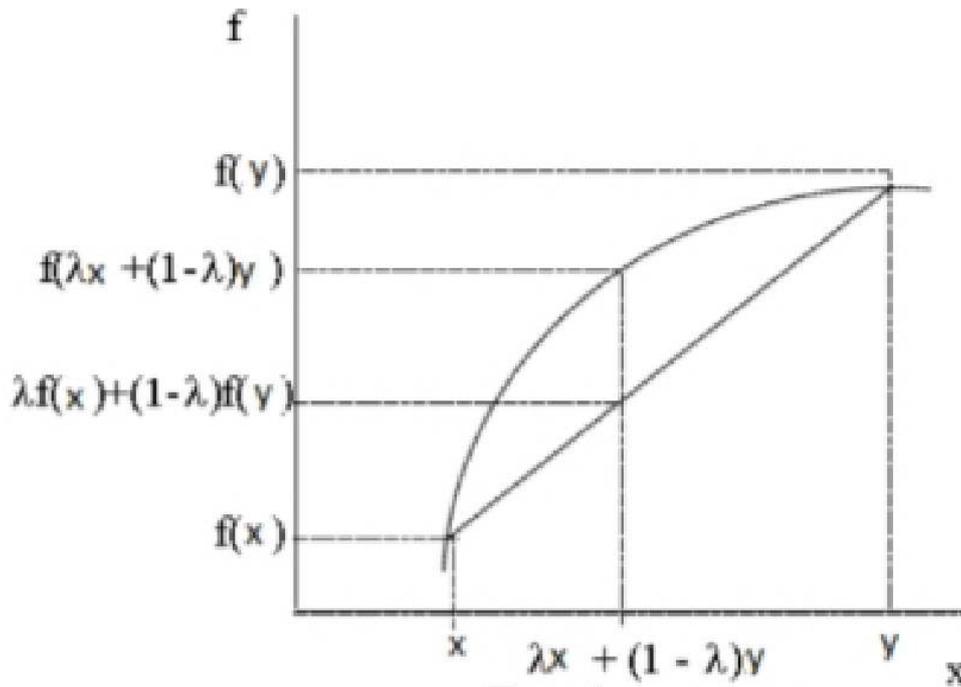


FIGURE 1.3: Fonction concave

Remarque 1.1. f est dite strictement convexe (respectivement strictement concave) si l'inégalité 1 (respectivement 1.2) stricte est toujours vérifiée pour $x \neq y$ et $\lambda \in]0, 1[$.

Propriété 1.1. .

1. f concave est équivalent à dire que $(-f)$ est convexe.
2. Si f et g sont deux fonctions concaves (respectivement convexes), alors

$$\forall (a, b) \in \mathbb{R}_+^2, \quad (af + bg),$$

est une fonction concave (respectivement convexe).

3. Si f est une fonction concave et g une fonction croissante et concave, alors $g(f(x))$ est concave.
4. Si f est une fonction convexe et g une fonction croissante et convexe, alors $g(f(x))$ est convexe.

Définition 1.9. (Fonction ni convexe ni concave)

Une fonction est dite ni convexe ni concave si elle ne satisfait ni la propriété de la convexité ni celle de la concavité sur tout son domaine.

Formellement, soit $f : I \rightarrow \mathbb{R}$ une fonction définie sur un intervalle I de \mathbb{R} . On dit que f est ni convexe ni concave si et seulement s'il existe au moins un intervalle J inclus dans I tel que pour tout couple de points (x, y) appartenant à J , le segment $[x, y]$ est strictement au-dessus ou strictement en dessous du graphe de f .

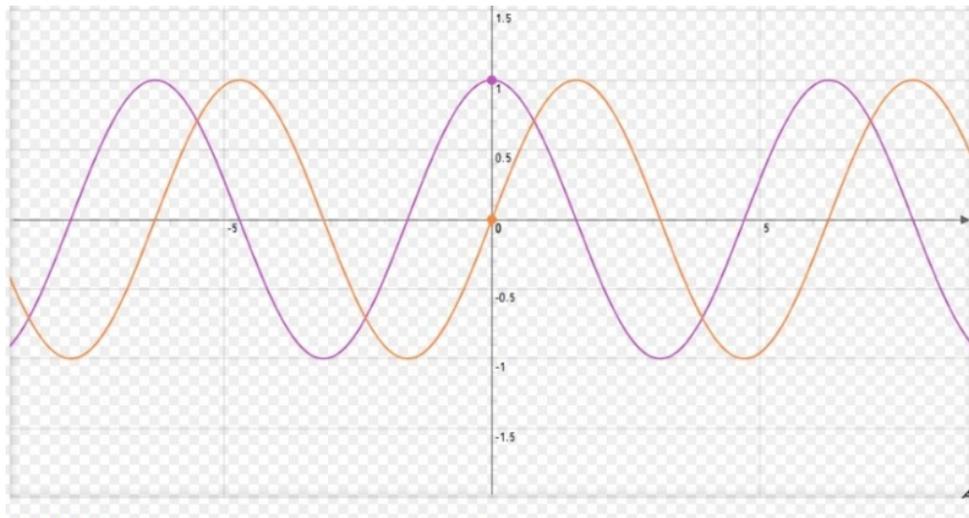


FIGURE 1.4: Fonction ni convexe ni concave

1.1.3 Ensemble et fonction affine

Définition 1.10. (*Ensemble affine*)

Un sous-ensemble $S \subseteq \mathbb{R}^n$ est dit affine si :

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

Définition 1.11. (*Fonction affine*)

Une fonction $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ est dite affine sur S si elle vérifié :

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

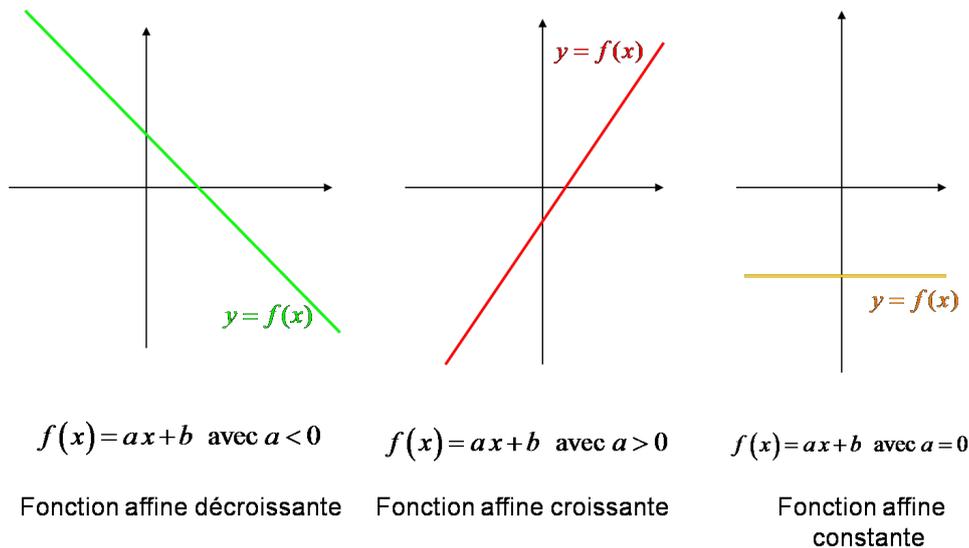


FIGURE 1.5: Fonction affine

Si de plus $f(0) = 0$ alors f est linéaire.

Définition 1.12. (*hyperplan*)

On appelle hyperplan dans \mathbb{R}^n un ensemble de la forme :

$$\pi = \{x \in \mathbb{R}^n / \langle c, x \rangle = \alpha\}.$$

Où $c \neq 0$

Dans l'espace \mathbb{R}^n ; un hyperplan défini deux demi-espaces

$$\{x \in \mathbb{R}^n / \langle c, x \rangle \leq \alpha\}, \quad \text{et} \quad \{x \in \mathbb{R}^n / \langle c, x \rangle \geq \alpha\}.$$

Définition 1.13. (Polyédre)

Un ensemble $S \subseteq \mathbb{R}^n$ est un polyédre si et seulement si :

$$S = \{x \in \mathbb{R}^n / Ax \geq b, \quad A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m\}.$$

1.1.4 Matrices et vecteurs

Dans cette partie, on présente quelques résultats connus sur les normes et les matrices.

Produit scalaire et norme

Définition 1.14. Le produit scalaire usuel de deux vecteurs x et y de \mathbb{R}^n est définie par :

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i = x^t y.$$

Définition 1.15. La norme vectorielle est une application de \mathbb{R}^n dans \mathbb{R}_+ notée par $\|\cdot\|$ et vérifié les conditions suivantes :

1. $\forall x \in \mathbb{R}^n : \|x\| = 0 \Leftrightarrow x = 0,$
2. $\forall \lambda \in \mathbb{R}, \forall x \in \mathbb{R}^n : \|\lambda x\| = |\lambda| \|x\|,$
3. $\forall x, y \in \mathbb{R}^n : \|x + y\| \leq \|x\| + \|y\|.$

Les normes usuelles sur \mathbb{R}^n , pour tout $x \in \mathbb{R}^n$ sont :

$$\begin{aligned} \|x\|_1 &= \sum_{i=1}^n |x_i|, \\ \|x\|_2 &= \sqrt{\sum_{i=1}^n (x_i)^2}, \\ \|x\|_\infty &= \max |x_i|. \end{aligned}$$

On note que \mathbb{R}^n les trois normes $\|x\|_1, \|x\|_2, \|x\|_\infty$

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq n \|x\|_\infty.$$

Définition 1.16. L'ensemble de m vecteurs v_i est dit linéairement indépendantes si et seulement si :

$$\sum_{i=1}^m \lambda_i v_i = 0 \Rightarrow \lambda_i = 0, \forall i = 1, \dots, m.$$

Remarque 1.2. Notons que la matrice $B \in \mathbb{R}^{m \times n}$ qui est formé de m vecteurs indépendantes est inversibles.

1.1.5 Matrices

Définition 1.17. Une matrice de taille $m \times n$ à coefficients dans \mathbb{R} est une famille de nombres $a_{ij} \in \mathbb{R}$ avec $1 \leq i \leq m$ et $1 \leq j \leq n$. Elle se dispose sous la forme rectangle

$$A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

- L'indice i est appelé l'indice ligne, l'indice j est appelé l'indice colonne.
- Si $m = n$, on note $M_n(\mathbb{R}) = M_{nn}(\mathbb{R})$ et on parle de matrices carrés d'ordre n .

Définition 1.18. (La transposée d'une matrice)

La transposée d'une matrice $A \in \mathbb{R}^{m \times n}$, que l'on note A^t , est la matrice de taille $n \times m$ telle que l'élément (j, i) est donné par a_{ij} pour $i = 1, \dots, m$ et $j = 1, \dots, n$. Autrement dit, la i -ième colonne de A^t vient de la i -ième ligne de A .

Propriété 1.2. On a les propriétés suivantes :

- $(A^t)^t = A$.
- Si λ est un scalaire, alors $(\lambda A)^t = \lambda A^t$.
- $(A^{-1})^t = (A^t)^{-1}$.
- $(A + B)^t = A^t + B^t$ Si A et B sont de la même taille.
- Si $A \in \mathbb{R}^{m \times p}$ et $B \in \mathbb{R}^{p \times n}$, alors $(AB)^t = B^t A^t$.

Remarque 1.3. Pour toute matrice A , le produit $A^t A$ est une matrice carrée symétrique et les éléments de sa diagonale principale sont non négatifs.

Définition 1.19. Soit $A \in \mathbb{R}^{m \times n}$ une matrice. Le nombre maximale des lignes (ou des colonnes) de A linéairement indépendantes est appelé le rang de A est noté par $\text{rg}(A)$.

De plus, A est dite de plein rang si :

$$\text{rg}(A) = \min(m, n).$$

Propriété 1.3. .

1. $\text{rg}(A^t) = \text{rg}(A)$.
2. Si le produit matriciel AB est défini, alors $\text{rg}(AB) \leq \min\{\text{rg}(A); \text{rg}(B)\}$.

Définition 1.20. 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 1.21. (Le gradient)

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

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

Définition 1.22. (la matrice hessienne)

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

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

est appelée matrice Hessienne de f en a .

Définition 1.23. (la matrice Jacobéenne)

Soit $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, La fonction $JF(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ est appelée matrice Jacobéenne et est définie par :

$$JF(x) = \nabla F(x)^t = \begin{pmatrix} \nabla f_1(x)^t \\ \vdots \\ \nabla f_m(x)^t \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

Remarque 1.4.

La matrice Hessienne est toujours symétrique.

1.1.6 Convexité et dérivée

Définition 1.24. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois continûment différentiable sur un domaine convexe S .

— f est une fonction convexe sur S si et seulement si la matrice hessienne est semi-définie positive c'est à dire :

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

— f est une fonction strictement convexe sur S si et seulement si la matrice hessienne est définie positive c'est à dire :

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

1.2 Programmation mathématique

1.2.1 Définitions

- **Problème d'optimisation**

Sous sa forme générale, un problème d'optimisation s'écrit comme suit :

$$\begin{cases} \min f(x) \\ x \in C, \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continue, $\emptyset \neq C \subseteq \mathbb{R}^n$ est l'ensemble des contraintes

Si $C = \mathbb{R}^n$, ce problème est appelé problème d'optimisation sans contraintes.

- **Programme mathématique**

Un Programme mathématique est un problème d'optimisation qui consiste à trouver une solution du problème qui maximise ou minimise une fonction donnée sous un ensemble des contraintes.

En générale, un programme mathématique est défini par :

$$(PM) \quad \begin{cases} \min f(x) \\ x \in D, \end{cases} .$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction objective de (PM), $D \subseteq \mathbb{R}^n$ est l'ensemble des contraintes.

Souvent D est présenté comme suit :

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

avec g_i, h_j sont des fonctions de $\mathbb{R}^n \rightarrow \mathbb{R}$.

- **Contrainte saturée(active)** : Une contrainte d'inégalité $g_i(x) \leq 0, \forall i$, est dite saturée ou (active) en $\bar{x} \in D$ si $g_i(\bar{x}) = 0$, pour tout $i = 1, \dots, m$.

Une contrainte d'égalité $h_j(x) = 0$ est par définition saturée en tout point x de D .

- **Solution réalisable(faisable)** : Un point x^0 de D est appelé solution réalisable de (PM) s'il vérifié les containtes c-à-d :

$$\exists x^0 \in \mathbb{R}^n : g_i(x) \leq 0; h_j(x_0) = 0; \quad \forall i = 1, \dots, m; \quad \forall j = 1, \dots, p.$$

- **Solution optimale globale** : Tout point $x^* \in D$ vérifié :

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

est appelée solution optimale globale

- **Solution optimale locale** : Tout point $x^* \in D$ vérifié :

$$f(x^*) \leq f(x), \forall x \in D \cap v,$$

où v est un voisinage de x^* est appelée solution optimale locale.

On note que $f(x^*)$ est appelée valeur optimale de (PM).

1.2.2 Classification d'un programme mathématique

La classification de (PM) est établis à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité de la fonction objectif et les contraintes :

1. (PM) est différentiable si les fonctions f, g_i, h_j sont différentiables.
2. (PM) est convexe si f, g_i sont convexes et h_j sont affines.

Remarque 1.5. Si (PM) convexe, alors tout optimum locale est un optimum globale.

1.2.3 Qualification des contraintes

La Qualification des contraintes est satisfaite pour tout $\bar{x} \in D$ dans l'un des trois cas suivantes :

- Les contraintes sont affines (D est un polyédre convexe).
- La condition de Slater : Si D est convexe (g_i sont convexes et h_j sont affines) et $\text{int}(D) \neq \emptyset$ c-à-d :

$$\exists x^0 : g_i(x^0) < 0 \quad \text{et} \quad h_j(x^0) = 0, \quad \forall i, j.$$

- La condition de Magazarien-Fromowitz : si les gradients des contraintes saturées en $\bar{x} \in D$ sont linéairement indépendantes.

1.2.4 Existence et unicité

Théorème 1.1. (Weirstrass) Si f est une fonction continue sur $D \subseteq \mathbb{R}^n$, D est compact (fermé et borné), alors (PM) admet au moins une solution optimale $x^* \in D$.

Corollaire 1.1. Si $D \subseteq \mathbb{R}^n$ est non vide et fermé et si f est continue et coercive sur D (c-à-d $f(x) \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$), alors (PM) admet au moins une solution optimale.

Si f est strictement convexe et l'ensemble D est convexe, alors (PM) admet une solution optimale, la solution est unique.

Remarque 1.6. La stricte convexité assure l'unicité de la solution et non l'existence de ce dernier.

1.2.5 Conditions d'optimalités

Le théorème suivant dit de Karush-kuhn-Tucker-Lagrange donne une condition nécessaire d'optimalité (condition du premier ordre).

Théorème 1.2. Supposons que les fonctions f , g_i , h_j sont C^1 dans un voisinage de $\bar{x} \in D$ et que les contraintes vérifient une des trois conditions de qualification ci-dessus. Si f a un minimum local en \bar{x} sur D alors $\exists \lambda \in \mathbb{R}_+^m$, $\exists \mu \in \mathbb{R}^p$ tel que :

$$(K.K.T) \quad \begin{cases} \nabla f(\bar{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\bar{x}) + \sum_{j=1}^p \mu_j \nabla h_j(\bar{x}) = 0, \\ \lambda_i g_i(\bar{x}) = 0, i = 1, \dots, m, \\ h_j(\bar{x}) = 0, i = 1, \dots, p. \end{cases}$$

les quantités λ_i et μ_j sont appelées multiplicateurs de Karush-kuhn-Tucker-Lagrange.

Remarque 1.7. 1. Si les contraintes de (PM) ne sont pas qualifiées en \bar{x} , les conditions de (K.K.T) ne s'appliquent pas.

2. Si (PM) est convexe, les conditions de (K.K.T) sont à la fois nécessaires et suffisantes pour que \bar{x} soit un minimum globale.

1.3 Méthode de Newton

Dans cette section, on s'intéresse par la méthode de Newton.

Cas particulier

On commence par un cas particulier : pour $f : \mathbb{R} \rightarrow \mathbb{R}$. On cherche $x \in \mathbb{R}$ tel que $f(x) = 0$.

Théorème 1.3. Soit $f : [a; b] \rightarrow \mathbb{R}$ une fonction de classe C^2 sur $[a, b]$, avec $f(a) > 0, f(b) < 0, f'$ et f'' gardent le même signe sur $[a, b]$ sans s'annuler.

Alors l'équation $f(x) = 0$ possède une unique racine c dans $[a, b]$

On utilise le développement de Taylor au voisinage de point x^k .

$$f(x) \approx f(x^k) + f'(x^k)(x - x^k).$$

On a : $f(x) = 0$ alors :

$$0 = f(x^k) + f'(x^k)(x - x^k).$$

qui équivalent à

$$-f(x^k) = f'(x^k)(x - x^k).$$

Si $f'(x^k) \neq 0$, donc

$$x - x^k = -\frac{f(x^k)}{f'(x^k)},$$

ce qui implique que

$$x = x^k - \frac{f(x^k)}{f'(x^k)}.$$

On pose $x = x^{k+1}$, on obtient :

$$x^{(k+1)} = x^{(k)} - \frac{f(x^k)}{f'(x^k)}.$$

On obtient alors le processus suivant :

$$\begin{cases} x^0 \text{ point initial} \\ x^{(k+1)} = x^{(k)} - \frac{f(x^k)}{f'(x^k)} \end{cases} \quad k > 0.$$

C'est la formule de la méthode de Newton.

1.3.1 Interprétation graphique

La méthode de Newton consiste à remplacer la courbe par sa tangente en une de ses deux extrémités. Le point x^1 est l'intersection de cette tangente avec l'axe (Ox).

Pour faire le dessin, on prend $x^0 = b$ tel que

$$f' > 0, f'' < 0 \quad \text{et} \quad f(b)f''(b) > 0$$

Traçons la tangente à la courbe représentative de f passant par $(b, f(b))$. L'équation de cette tangente est

$$y = f'(b)(x - b) + f(b).$$

Son intersection avec l'axe (Ox) a une ordonnée nulle et son abscisse vaut

$$x^1 = b - \frac{f(b)}{f'(b)}.$$

On trace ensuite la tangente à la courbe au point $(x^1, f(x^1))$. Le réel x_2 est l'abscisse de l'intersection de cette deuxième tangente avec l'axe (Ox) et on réitère ce procédé.

Remarque 1.8. Si on prenait la tangente à la courbe en $(a, f(a))$, son intersection avec l'axe (Ox) ne serait pas, sur ce dessin, dans l'intervalle $[a, b]$

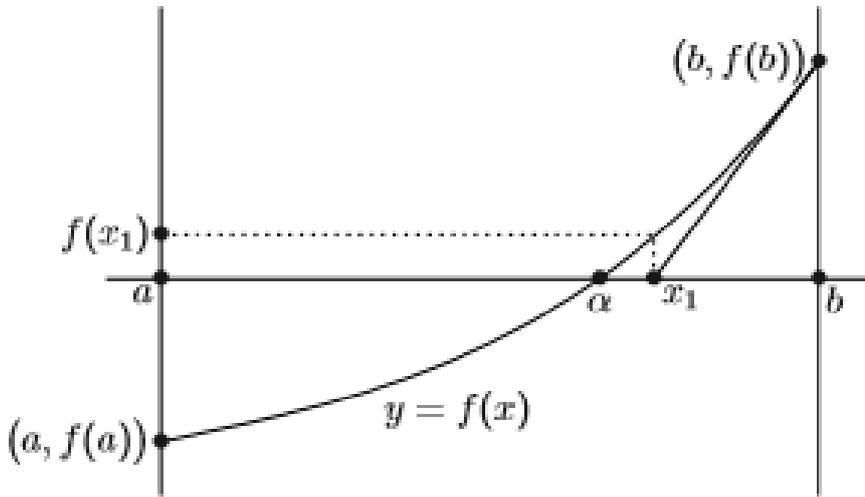


FIGURE 1.6: Interprétation graphique de la méthode de Newton

Remarque 1.9.

Si on prenait la tangente à la courbe en $(a, f(a))$, son intersection avec l'axe (Ox) ne serait pas, sur ce dessin, dans l'intervalle $[a, b]$.

Cas générale

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, on cherche $x^* \in \mathbb{R}^n$ tel que $f(x) = 0$.

Autrement dit, soit le système d'équations non linéaires sous la forme suivante :

$$\begin{cases} f_1(\bar{x}_1, \dots, \bar{x}_n) = 0, \\ f_2(\bar{x}_1, \dots, \bar{x}_n) = 0, \\ \dots \\ f_n(\bar{x}_1, \dots, \bar{x}_n) = 0. \end{cases}$$

Chaque équation $f_i(x_1, \dots, x_n) = 0$ est considéré comme une surface de $\mathbb{R}^n \times \mathbb{R}^n$, l'intersection de n surfaces nous donne la solution $\bar{x} \in \mathbb{R}^n$, tel que $F(\bar{x}) = 0$. Alors on peut remplacer chaque surface associé à $f_i(x) = 0$ par l'hyperplan qui est tangent au point $x^{(k)}$.

On définit alors le point $x^{(k+1)}$ comme l'intersection de ces n hyperplan avec l'hyperplan $y = 0$. on a

$$F(x^{(k+1)}) \simeq F(x^{(k)}) + F'(x^{(k)})(x^{(k+1)} - x^{(k)}),$$

alors :

$$0 = F(x^{(k)}) + F'(x^{(k)})(x^{(k+1)} - x^{(k)}),$$

et par conséquent :

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}).$$

Puisque $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, alors $F'(x^{(k)}) = J(x^{(k)})$ ou $J(x)$ désigne la matrice jacobienne donc :

$$x^{(k+1)} = x^{(k)} - [J(x^{(k)})]^{-1} F(x^{(k)}).$$

Pour éviter le calcul de l'inverse de la matrice jacobienne, on résoud le système linéaire suivant :

$$F'(x^{(k)})\Delta x^{(k)} = -F(x^{(k)}),$$

on prend :

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}.$$

ou Δx^k désigne la dérivation obtenue par la méthode de newton et x^{k+1} le nouveau itéré .

2

Méthode de Trajectoire Centrale avec poids pour PL

Dans ce chapitre, on s'intéresse à la méthode de points intérieurs de trajectoire centrale (TC) de type primal-dual basée sur une nouvelle direction de recherche pour résoudre un programme linéaire. On note que l'algorithme associé à cette étude théorique est déjà présenté par Darvay en 2002 (voir [6]).

2.1 Position de problème

Problème primal

On considère le programme linéaire sous la forme primal standard suivants

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

où A une matrice de type $(m \times n)$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et x vecteur inconnu de \mathbb{R}_+^n .

On notera par la suite :

$$\mathcal{F}_p = \{x \in \mathbb{R}_+^n, \quad Ax = b, \geq 0\}.$$

$$\mathcal{F}_p^0 = \{x \in \mathbb{R}_{++}^n, \quad Ax = b, > 0\}.$$

L'ensemble des solutions réalisable (et strictement réalisable, respectivement) pour (P).

La valeur optimal primale du problème est définie par :

$$Val(P) = \inf_x \{c^t x : Ax = b, \quad x \in \mathbb{R}_+^n\}.$$

On dit que x^* est un solution optimal primale de (P) si :

$$x^* \in \mathcal{F}_p \quad \text{et} \quad Val(P) = c^t x^*.$$

Problème dual

Pour obtenir le problème dual de (P), On considéré la fonction **lagrangienne** suivante :

$$L : \mathbb{R}_+^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$$

$$L(x, y) = c^t x + (b - Ax)^t y.$$

Le dual de (P) est donné par :

$$\max_{y \in \mathbb{R}^m} \left(\min_{x \geq 0} L(x, y) \right) = \max_{y \in \mathbb{R}^m} H(y),$$

où

$$\begin{aligned} H(y) &= \min_{x \in \mathbb{R}_+^n} L(x, y), \quad y \in \mathbb{R}^m, \\ &= \min_{x \in \mathbb{R}_+^n} [c^t x + (b - Ax)^t y], \\ &= \min_{x \in \mathbb{R}_+^n} [c^t x + b^t y - x^t A^t y], \\ &= \min_{x \in \mathbb{R}_+^n} [b^t y + (c - A^t y)x], \\ H(y) &= \begin{cases} b^t y & \text{si } c - A^t y \geq 0, \\ -\infty & \text{sinon.} \end{cases} \end{aligned}$$

Donc, le dual du programme (P) est donnée par :

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

Où l'inconnu est $(y, z) \in \mathbb{R}^m \times \mathbb{R}_+^n$.

On note par la suite

$$\mathcal{F}_D = \{z \in \mathbb{R}_+^n \text{ et } y \in \mathbb{R}^m : A^t y + z = c\},$$

$$\mathcal{F}_D^0 = \{z \in \mathbb{R}_{++}^n \text{ et } y \in \mathbb{R}^m : A^t y + z = c\}.$$

L'ensemble des solutions réalisable (et strictement réalisable respectivement) pour (D).

La valeur optimale dual du problème (D) est définie par :

$$Val(D) = \sup_{(y, z)} \{b^t y, A^t y + z = c, \quad y \in \mathbb{R}^m \text{ et } z \in \mathbb{R}_+^n\}.$$

On dit que (y^0, z^0) est une solution optimal dual de (D) si :

$$(y^0, z^0) \in \mathcal{F}_D \text{ et } Val(D) = b^t y^0.$$

2.1.1 Dualité faible

Définition 2.1. (Saut de dualité)

Soient $x \in F_P$, $(y, z) \in F_D$, alors la différence :

$$\begin{aligned} val(P) - val(D) &= c^t x - b^t y, \\ &= x^t z, \end{aligned}$$

est appelée le saut de dualité des problèmes (P) et (D).

Théorème 2.1. (Proposition [19]) Soit x et z deux solutions réalisables de (P) et (D) respectivement, alors

$$c^t x - b^t y = x^t z \geq 0.$$

Si le saut de dualité $x^t z = 0$, alors x est un solution optimal de (P) et (y, z) solution optimale de (D) .

Preuve.

On a :

$$\begin{aligned} c^t x - b^t y &= (A^t y + z)^t x - (Ax)^t y, \\ &= (y^t A + z^t) x - x^t A^t y, \\ &= \langle A^t y, x \rangle + \langle z, x \rangle - \langle Ax, y \rangle. \end{aligned}$$

Comme $\langle A^t y, x \rangle = \langle Ax, y \rangle$, donc :

$$\begin{aligned} c^t x - b^t y &= x^t z, \\ &= \sum_{i=1}^n x_i z_i \geq 0 \text{ car } x \geq 0 \text{ et } z \geq 0. \end{aligned}$$

Ce qui achève la preuve. □

2.1.2 Dualité forte

Théorème 2.2. [19]

Soient $(x, y) \in F_P \times F_D$ une solution réalisable de (P) et (D) , respectivement tel que :

$$b^t y = c^t x.$$

Alors x et y sont des solutions optimales de (P) et (D) .

Corollaire 2.1.

- Si $c^t x$ est non borné inférieurement sur F_P alors (D) est non réalisable (l'ensemble des contraintes duales F_D est vide).
- Si $b^t y$ est non borné supérieurement alors (P) est non réalisable (c-à-d : F_P est vide).
- Si (P) est non réalisable alors (D) est non borné (et réciproquement).

2.2 Modélisation de quelques problèmes réels sous forme d'un PL

Rappel

Les étapes de modélisation d'un problème de PL sont les suivantes :

- 1) Identifier les variables de décision .
- 2) Identifier les contraintes en les exprimant sous la forme d'équations ou d'inéquations linéaires.
- 3) Identifier l'objectif et le représenter sous forme linéaire en fonction des variables de décision, puis spécifier s'il faut maximiser ou bien minimiser l'objectif.

Exemple 2.1. Le problème de la nourriture pour chiens

Imaginons la situation suivante :

- Soit une entreprise de nourriture pour chiens, qui fabrique 2 types de granulés : le Wag-Tail (W) et le Bark-Mad (B).
- Chacun des types utilise un mélange de légumes, bœuf et poisson, dans les proportions suivantes :

Ingrédient	Qté totale	Qté dans B	Qté dans W
Légumes	1400 kg	4 kg	4 kg
Poisson	1800 kg	6 kg	3 kg
Bœuf	1800 kg	2 kg	6 kg

- On suppose que la compagnie opère un bénéfice de 12 Euros sur chaque paquet de B et de 8 euros sur ceux de W .
- Comment l'entreprise peut-elle faire pour maximiser son profit ?

Formulation

On note qu'on cherche à maximiser une quantité en fonction de diverses contraintes. Il s'agit bien d'un problème d'optimisation. On peut exprimer le problème de la manière suivante :

- On note par B le nombre de paquet de Bark-Mad produits, et par W le nombre de paquets de Wag-Tail.
- De la table précédente on déduit que la quantité totale de légumes consommées sera de $4W+4B$ kg, mais qu'on ne peut en aucun cas utiliser plus de 1400kg de légumes. Donc :

$$4B + 4W \leq 1400$$

- De même, en interprétant les contraintes sur le poisson et le bœuf, on obtient :

$$6B + 3W \leq 1800$$

et

$$2B + 6W \leq 1800$$

- finalement B et W sont forcément tous deux positifs.
- Du point de vue de l'optimisation du profit, on peut écrire

$$P = 12B + 8W$$

On cherche à maximiser P , le profit donc, le (PL) de ce problème peut s'écrire sous la forme suivant :

$$\left\{ \begin{array}{l} \max P = 12B + 8W \\ 4B + 4W \leq 1400 \\ 6B + 3W \leq 1800 \\ 2B + 6W \leq 1800 \\ B, W \geq 0 \end{array} \right.$$

Exemple 2.2. Problème de régime

- On veut suivre un régime qui impose de manger un élément des 4 groupes de bases : chocolat, crème glacée, soda et gâteau.
- Une barre de chocolat coûte 50 centimes, une part de crème glacée 20 centimes, chaque bouteille de soda 30 centimes et une part de gâteau 80 centimes.

- Chaque jour je dois ingérer 500 calories, 60g de chocolat, 100g de sucre et 80g de lipides.
- Le contenu nutritinnel de chaque type de nourriture est donné ainsi

	Cal.	Choc.(g)	Sucr.(g)	Lip.(g)
barre chocolat	400	30	20	20
crème glacée	200	20	20	40
cola	150	0	40	10
gâteau	500	0	40	50

On veut minimizer le coût du régime donc le (PL) peut s'écrire comme suit

$$\left\{ \begin{array}{l} \min z = 50x_1 + 20x_2 + 30x_3 + 80x_4 \quad (\text{Fonction objectif}) \\ 400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500 \quad (\text{Total calories}) \\ 30x_1 + 20x_2 \geq 60 \quad (\text{Total chocolat}) \\ 20x_1 + 20x_2 + 40x_3 + 40x_4 \geq 100 \quad (\text{Total sucres}) \\ 20x_1 + 40x_2 + 10x_3 + 50x_4 \geq 80 \quad (\text{Total lipides}) \\ x_i \geq 0, i = 1, 2, \dots, 4. \end{array} \right.$$

Exemple 2.3.

Une avion peut porter W kg supplémentaire sur un vol régulier, il y a n colis à transporter et chaque colis $i, i = 1, 2, \dots, n$ pèse a_i et génère un profit de p_i en dinar.

Ce problème est un modèle du sac à dos.

Considérons les variables binaires

$$x_i = \begin{cases} 1 & \text{si le coli } i, i = 1, 2, 3, \dots, n \\ 0 & \text{si non .} \end{cases}$$

Le modèle peut s'écrire sous la forme suivante :

$$\left\{ \begin{array}{l} \max Z = \sum_{i=1}^n p_i x_i \\ \sum_{i=1}^n a_i x_i \leq W, \\ x \in \{0, 1\}. \end{array} \right.$$

2.3 Méthode 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 : une complexité polynomiale et une convergence linéaire.

On rappelle qu'on cherche à résoudre un programme linéaire (P) qui est définie par :

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

Et son dual est donné par :

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

Pour résoudre les deux problèmes (P) et (D) on suppose qu'ils vérifient les hypothèses suivantes :

Hypothèses

1. la matrice A est de plein rang (c-à-d : $rg(A) = m \leq n$).
2. Condition de point intérieur (CPI) : $\exists(x^0, y^0, z^0) \in F_P^+ \times F_D^+$.

$$\begin{cases} Ax^0 = b, \\ A^t y^0 + z^0 = c, \\ x^0 > 0, z^0 > 0, y \in \mathbb{R}^m. \end{cases}$$

On note que (P) consiste à minimiser une fonction linéaire (convexe) sous l'ensemble des contraintes qui est un polyèdre convexe. Comme F_P est non vide (hypothèse 2), alors $\exists z \in \mathbb{R}_+^n$; $\exists y \in \mathbb{R}^m$. D'après les conditions d'optimalité de Karuch-Kuhn-Tucher :

$$(K.K.T) \begin{cases} \nabla(c^t x) - \sum_{i=1}^n z_i (\nabla x_i) + \sum_{j=1}^m y_j (b - Ax) = 0, \\ Ax = b, \\ zx = 0. \end{cases}$$

Après la dérivation de premier équation de ce système on obtient :

$$\begin{cases} c - z - A^t y = 0, \\ Ax = b, \\ zx = 0. \end{cases}$$

Donc :

$$\begin{cases} Ax = b, \\ A^t y + z = c, \\ xz = 0, x > 0, z > 0, y \in \mathbb{R}^m. \end{cases} \tag{2.1}$$

Dont la dernière équation est dite équation de complémentarité.

L'idée principale de la méthode de trajectoire centrale est de perturbé la dernière équation du système (2.1). Alors

$$\begin{cases} Ax = b, \\ A^t y + z = c, \\ xz = \mu e, x > 0, z > 0, y \in \mathbb{R}^m. \end{cases} \tag{2.2}$$

Tel que : $\mu > 0, e = (1, 1, \dots, 1)^t \in \mathbb{R}^n$.

L'ensemble :

$$\{x(\mu), y(\mu), z(\mu); \mu > 0\},$$

s'appelle la trajectoire centrale. De plus ; si $\mu \rightarrow 0$ alors $x^t z$ tend vers le zéro c-à-d trouver une solution optimale pour (P) et (D) revient à résoudre le système (2.2).

Le système (2.2) devient :

$$\begin{cases} Ax - b = 0, \\ A^t y + z - c = 0, \\ xz - \mu e = 0. \end{cases} \tag{2.3}$$

Ce système est un système d'équation non linéaire. Il est résolu par la méthode de Newton appliquée à la fonction

$$F(x, y, z) = 0,$$

où $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$ définie par :

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

Soient $(x, y, z) \in F_p^+ \times F_D^+$, tel que $xz \neq \mu e$.

La direction de Newton $(\Delta x, \Delta y, \Delta z)$ est la solution de système linéaire :

$$\nabla F(x, y, z) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F(x, y, z).$$

On va calculer la matrice jacobienne de F :

$$\nabla F(x, y, z) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I_n \\ z & 0 & x \end{pmatrix}.$$

Ce qui est équivalent à résoudre le système linéaire suivant :

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

On défini le vecteur v pour tout vecteur réalisable primal $x > 0$; et le vecteur réalisable dual $z > 0$,

$$v = \sqrt{\frac{xz}{\mu}}.$$

Proposition 2.1. On a

$$\Delta x^t \Delta z = 0.$$

Preuve. D'après l'équation (2) de système (2.4) on à :

$$\Delta z = -A^t \Delta y,$$

Donc

$$\begin{aligned} \Delta x^t \Delta z &= \Delta x^t (-A^t \Delta y), \\ &= -(\Delta x^t A^t) \Delta y, \\ &= -(A\Delta x)^t \Delta y. \end{aligned}$$

D'après l'équation (1) de système (2.4) on a :

$$A\Delta x = 0,$$

alors

$$\Delta x^t \Delta z = 0.$$

□

2.3.1 La mesure de proximité

Dans cette partie, on est obligé de définir une mesure de proximité pour que le nouveau itéré produit par l'algorithme reste réalisable et proche de la trajectoire centrale.

Donc :

$$\delta = \delta(x, z, \mu) = \|\mu e - xz\|.$$

C'est facile de vérifier que :

$$\delta = 0 \Leftrightarrow xz = \mu e.$$

2.3.2 La description de l'algorithme

- On commence par un point (x^0, y^0, z^0) qui vérifie la condition de point intérieur avec $x^0 > 0$, $z^0 > 0$ et $\delta(x^0, z^0; \mu^0) < \beta$; où μ^0 connu et $0 < \beta < 1$.
- On calcule le saut de dualité ; s'il est inférieur à ϵ alors on obtient la solution optimale si non,
- on réduit μ par un facteur de $(1 - \theta)$; où $0 < \theta < 1$ c-à-d :

$$\mu^+ = (1 - \theta)\mu,$$

- on calcule les directions $(\Delta x, \Delta y, \Delta z)$ par la résolution du système linéaire (2.4).
- Le nouveau itéré obtenu après le pas de Newton complet par :

$$\begin{cases} x^+ = x + \Delta x, \\ y^+ = y + \Delta y, \\ z^+ = z + \Delta z. \end{cases}$$

- On répète la procédure jusqu'à ce que le saut de dualité tend vers le zéro.
- Dans ce cas, on obtient une solution optimale approximative.

Maintenant, on présente un algorithme primal-dual à petit pas pour tracer approximativement la trajectoire centrale.

2.3.3 Algorithme TC classique

Dans cette partie, on présente un algorithme de trajectoire central de type primal-dual pour un (PL) :

Algorithm 1 Algorithme primal-dual à petit pas pour PL**Début d'algorithme****Données :**

un paramètre de précision $\epsilon > 0$;

un paramètre de proximité $0 < \beta < 1$ (défaut $\beta = \frac{1}{\sqrt{2}}$) ;

un paramètre $0 < \theta < 1$ (défaut $\theta = \frac{1}{\sqrt{2n}}$)

Initialisation : soit (x^0, y^0, z^0) vérifiant la **CPI** tel que :

$$\mu^{(0)} = \frac{(x^0)^t z^0}{n} \quad \text{et} \quad \delta(x^0, z^0; \mu^{(0)}) \leq \beta \quad \text{et} \quad k = 0 ;$$

Tant que $n\mu^{(k)} \geq \epsilon$ faire :

- $\mu^{(k+1)} = (1 - \theta)\mu^{(k)}$.
- calculer $(\Delta x^{(k)}, \Delta y^{(k)}, \Delta z^{(k)})$ via le système (2.4) ;
- mise à jour $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}, y^{(k+1)} = y^{(k)} + \Delta y^{(k)}, z^{(k+1)} = z^{(k)} + \Delta z^{(k)}$ et $k = k + 1$;

Fin Tant que

Fin algorithme.

On note que Roos et all en 1997 [19] ont montré que l'algorithme converge durant un temps d'exécution minimal ainsi, le nombre des itérations qui sera minimal et si la solution initiale est réalisable et $x^0, z^0 > 0$ proche de la trajectoire centrale avec $\theta = \frac{1}{\sqrt{2n}}$ et $\beta = \frac{1}{\sqrt{2}}$ alors la complexité de l'algorithme est donné par le théorème suivant :

Théorème 2.3 (Roos [19]). Soient $\theta = \frac{1}{\sqrt{2n}}$, $\beta = \frac{1}{\sqrt{2}}$. Alors, l'algorithme de la trajectoire centrale de type primal-dual avec le pas de Newton complet nécessite de $O\left(\sqrt{n} \log\left(\frac{(x^0)^t z^0}{\epsilon}\right)\right)$ itérations.

2.4 Méthode de TC avec Poids

On rappelle que la TC est l'ensemble des solutions

$$\{x(\mu), y(\mu), z(\mu); \mu > 0\},$$

de système non linéaire (2.2). L'idée principale de la méthode trajectoire centale avec poids est de remplacer la dernière équation du système (2.2) par l'équation paramétrisé $xz = w^2$ tel que w est un vecteur positif arbitraire. Donc, on obtient le système suivant.

$$\begin{cases} Ax = b, \\ A^t y + z = c, \\ xz = w^2, \\ x \geq 0, z \geq 0, y \in \mathbb{R}^m. \end{cases} \quad (2.5)$$

où $w > 0$. Si la condition de point intérieur est satisfaite, alors le système (2.5) possède une solution unique. Cette caractéristique a été prouvée pour la première fois par Kojima et al [13]. On peut donc appliquer la méthode de Newton pour le système (2.5) afin de développer un algorithme primal-dual de TC avec poids.

Dans la partie suivante, on présente une nouvelle méthode pour trouver des directions de recherche en appliquant la méthode de Newton pour une forme équivalente du système (2.5).

2.4.1 Nouvelle direction de recherche

Dans cette partie, on introduit une nouvelle méthode pour construire des directions de recherche en utilisant le système (2.5).

On considère la fonction

$$\varphi \in C^1, \quad \varphi : \mathbb{R}^+ \rightarrow \mathbb{R}^+.$$

De plus, on suppose que la fonction inverse φ^{-1} existe. Alors, le système (2.5) peut s'écrire sous la forme suivante :

$$\begin{cases} Ax &= b, & x \geq 0, \\ A^t y + z &= c, & z \geq 0, y \in \mathbb{R}^m, \\ \varphi(xz) &= \varphi(w^2), \end{cases} \quad (2.6)$$

et on peut appliquer la méthode de Newton pour le système (2.6) pour obtenir une nouvelle classe de directions de recherche.

On suppose que on a

$$Ax = b$$

et

$$A^t y + z = c,$$

pour un triple (x, y, z) tel que $x > 0$ et $z > 0$, donc x et z sont strictement réalisables. En appliquant la méthode de Newton sur le système non linéaire (2.6), on obtient :

$$\begin{cases} A\Delta x = 0, \\ A^t \Delta y + \Delta z = 0, \\ z\varphi'(xz)\Delta x + x\varphi'(xz)\Delta z = \varphi(w^2) - \varphi(xz). \end{cases} \quad (2.7)$$

Pour les besoins théoriques, on définit :

$$v = \sqrt{xz} \quad \text{et} \quad d = \sqrt{xz^{-1}},$$

ce qui implique que :

$$d^{-1}x = dz = v.$$

De plus,

$$d_x = d^{-1}\Delta x, \quad d_z = d\Delta z,$$

et observe que on a

$$v(d_x + d_z) = z\Delta x + x\Delta z \quad (2.8)$$

et

$$d_x d_z = \Delta x \Delta z. \quad (2.9)$$

Par conséquent, le système linéaire (2.7) peut s'écrire sous la forme suivante :

$$\begin{cases} \bar{A}d_x = 0, \\ \bar{A}^t \Delta y + d_z = 0, \\ d_x + d_z = p_v, \end{cases} \quad (2.10)$$

où

$$p_v = \frac{\varphi(w^2) - \varphi(v^2)}{v\varphi'(v^2)}, \quad (2.11)$$

et $\bar{A} = \text{Adiag}(d)$, avec

$$\text{diag}(\xi) = \begin{bmatrix} \xi_1 & 0 & \dots & 0 \\ 0 & \xi_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \xi_n \end{bmatrix},$$

pour tout vecteur ξ . Dans la partie suivante, on développera un nouvel algorithme primal-dual de TC avec poids basé sur une nouvelle direction de recherche.

2.4.2 Algorithme

Dans cette partie, on pose

$$\varphi(t) = \sqrt{t},$$

et on développe un nouvel algorithme primal-dual de TC avec poids basé sur une nouvelle direction de recherche appropriées. Ainsi, en effectuant la substitution $\varphi(t) = \sqrt{t}$ dans (2.11), on obtient :

$$p_v = 2(w - v). \quad (2.12)$$

maintenant pour tout vecteur positif v , on définit la mesure de proximité comme suit :

$$\delta(x, z, w) = \frac{\|p_v\|}{2 \min(w)} = \frac{\|w - \sqrt{xz}\|}{\min(w)}, \quad (2.13)$$

où $\|\cdot\|$ est la norme euclidienne (l_2 norme).

On introduit une autre mesure :

$$\sigma_c(w) = \frac{\max(w^2)}{\min(w^2)},$$

dont le rôle de $\sigma_c(w)$ est de mesurer la distance de w^2 au trajectoire centrale. De plus, on pose :

$$q_v = d_x - d_z.$$

On note que

$$d_x^t d_z = 0,$$

ce qui implique que les vecteurs d_x et d_z sont orthogonaux. On trouve ainsi que

$$\|p_v\| = \|q_v\|.$$

Par conséquent, la mesure de proximité peut s'écrire comme suit :

$$\delta(x, z, w) = \frac{\|q_v\|}{2 \min(w)}. \quad (2.14)$$

Ainsi, on obtient :

$$d_x = \frac{p_v + q_v}{2}, \quad d_z = \frac{p_v - q_v}{2},$$

et

$$d_x d_z = \frac{p_v^2 - q_v^2}{4}. \quad (2.15)$$

En faisant la substitution $\varphi(t) = \sqrt{t}$ en (2.7) on obtient :

$$\begin{cases} A\Delta x = 0, \\ A^t \Delta y + \Delta s = 0, \\ \sqrt{\frac{z}{x}} \Delta x + \sqrt{\frac{x}{z}} \Delta z = 2(w - \sqrt{xz}). \end{cases} \quad (2.16)$$

Algorithme de TC avec Poids

Par la suite, on présente un algorithme réalisable de trajectoire central avec poids basé sur une nouvelle direction de recherche pour résoudre (P) et (D) approximativement.

Algorithm 2 : Algorithme de trajectoire central avec poids

Début d'algorithme

Données :

un paramètre de précision $\epsilon > 0$;

un paramètre de proximité $0 < \beta < 1$ (défaut $\beta = \frac{1}{2}$) ;

un paramètre $0 < \theta < 1$.

Initialisation : soit (x^0, y^0, z^0) vérifiant la CPI tel que $w^0 = \sqrt{x^0 z^0}$ et $\delta(x^0, z^0, w^0) < \beta$ et $k = 0$,

Tant que $(x^{(k)})^t z^{(k)} > \epsilon$ faire :

- $w^{(k+1)} = (1 - \theta)w^{(k)}$;
- Calculer $(\Delta x, \Delta y, \Delta z)$ via le système (2.16)
- Mise à jour $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$, $y^{(k+1)} = y^{(k)} + \Delta y^{(k)}$, $z^{(k+1)} = z^{(k)} + \Delta z^{(k)}$ et $k = k + 1$;

Fin Tant que

Fin algorithme.

On note que, Darvay et al [6] en 2002 ont prouvé que l'Algorithme 2.4.2 est bien défini et converge vers une solution optimale selon de choix précisé pour les valeurs de β et θ tel que le nombre d'itérations effectuées par l'algorithme soit minimale.

2.4.3 Convergence de l'algorithme

Analyse de la stricte de faisabilité

Le lemme suivant, nous donne une condition suffisante pour assuré la stricte de faisabilité de nouveau itéré

$$x^+ = x + \Delta x,$$

et

$$z^+ = z + \Delta z.$$

Obtennus après le pas de Newton complet.

Lemme 2.1. [[6], Lemme 4.1 en 2002] (condition suffisante)

Soit $\delta = \delta(x, z, w) < 1$. Alors le nouveau itéré obtenu après le pas de Newton complet est strictement réalisable, c'est à dire

$$x^+ > 0 \quad \text{et} \quad z^+ > 0.$$

Preuve. Pour tout $0 \leq \alpha \leq 1$ soit $x^+(\alpha) = x + \alpha \Delta x$ et $z^+(\alpha) = z + \alpha \Delta z$. Ainsi,

$$x^+(\alpha)z^+(\alpha) = xz + \alpha(z\Delta x + x\Delta z) + \alpha^2 \Delta x \Delta z.$$

En utilisant (2.8) et (2.9) on obtient

$$x^+(\alpha)z^+(\alpha) = v^2 + \alpha v(d_x + d_z) + \alpha^2 d_x d_z,$$

et à partir de(2.10) et (2.15), on obtient :

$$x^+(\alpha)z^+(\alpha) = (1 - \alpha)v^2 + \alpha(v^2 + vp_v) + \alpha^2\left(\frac{p_v^2}{4} - \frac{q_v^2}{4}\right).$$

De plus, l'équation (2.12) implique que

$$v + \frac{p_v}{2} = w,$$

et donc

$$v^2 + vp_v = w^2 - \frac{p_v^2}{4},$$

par conséquent

$$x^+(\alpha)z^+(\alpha) = (1 - \alpha)v^2 + \alpha\left(w^2 - (1 - \alpha)\frac{p_v^2}{4} - \alpha\frac{q_v^2}{4}\right). \quad (2.17)$$

Ainsi, l'inégalité $x^+(\alpha)z^+(\alpha) > 0$ est satisfaite si

$$\left\| (1 - \alpha)\frac{p_v^2}{4} + \alpha\frac{q_v^2}{4} \right\|_{\infty} < \min(w^2).$$

En utilisant(23) et (2.14), on obtient :

$$\begin{aligned} \left\| (1 - \alpha)\frac{p_v^2}{4} + \alpha\frac{q_v^2}{4} \right\|_{\infty} &\leq (1 - \alpha)\frac{\|p_v^2\|_{\infty}}{4} + \alpha\frac{\|q_v^2\|_{\infty}}{4}, \\ &\leq (1 - \alpha)\frac{\|p_v\|^2}{4} + \alpha\frac{\|q_v\|^2}{4}, \\ &= \delta^2 \min(w^2), \\ &< \min(w^2). \end{aligned}$$

Donc, pour tout $0 \leq \alpha \leq 1$ on a

$$x^+(\alpha)z^+(\alpha) > 0.$$

Par conséquent, on observe que les fonctions linéaires de α , $x^+(\alpha)$ et $z^+(\alpha)$ ne changent pas de signe sur l'intervalle $[0, 1]$. Pour $\alpha = 0$ on a $x^+(0) = x > 0$ et $z^+(0) = z > 0$, donc on obtient $x^+(1) = x^+ > 0$ et $z^+(1) = z^+ > 0$, et qui termine la preuve. \square

Convergence quadratique de la mesure de proximité

Le lemme suivant montre la convergence quadratique de la mesure de proximité.

Lemme 2.2. *[[6], Lemme 4.2 en 2002]*

Soient $x^+ = x + \Delta x$ et $z^+ = z + \Delta z$ les nouveaux itérés obtenus après le pas de Newton complet, on note par :

$$v^+ = \sqrt{x^+ z^+}.$$

Supposons que $\delta = \delta(x, z, w) < 1$. Alors

$$\delta(x^+, z^+, w) \leq \frac{\delta^2}{1 + \sqrt{1 - \delta^2}},$$

ainsi $\delta(x^+, z^+, w) < \delta^2$, ce qui montre la convergence quadratique de la méthode de Newton.

Preuve. D'après le lemme 2.1 on obtient $x^+ > 0$ et $z^+ > 0$. Maintenant, on remplace $\alpha = 1$ dans (2.17) et obtient :

$$(v^+)^2 = w^2 - \frac{q_v^2}{4}. \quad (2.18)$$

En utilisant (2.18), on obtient :

$$\begin{aligned} \min((v^+)^2) &\geq \min(w^2) - \frac{\|q_v^2\|_\infty}{4}, \\ &\geq \min(w^2) - \frac{\|q_v\|^2}{4}, \\ &= \min(w^2)(1 - \delta^2), \end{aligned}$$

et par conséquent

$$\min(v^+) \geq \min(w)\sqrt{1 - \delta^2}. \quad (2.19)$$

De plus, à partir les équations (2.18) et (19) on obtient :

$$\begin{aligned} \delta(x^+, z^+, w) &= \frac{1}{\min(w)} \left\| \frac{w^2 - (v^+)^2}{w + v^+} \right\|, \\ &\leq \frac{\|w^2 - (v^+)^2\|}{\min(w)(\min(w) + \min(v^+))}, \\ &\leq \frac{\|q_v^2\|}{(2\min(w))^2(1 + \sqrt{1 - \delta^2})}, \\ &\leq \frac{1}{1 + \sqrt{1 - \delta^2}} \left(\frac{\|q_v\|}{2\min(w)} \right)^2, \\ &= \frac{\delta^2}{1 + \sqrt{1 - \delta^2}}. \end{aligned}$$

Par conséquent, on a $\delta(x^+, z^+, w) < \delta^2$, ce qui termine la preuve. \square

L'influence du nouveau itéré en saut de dualité

la lemme suivant, nous donne une borne supérieure pour le saut de dualité obtenu après le pas de Newton complet.

Lemme 2.3. [[6], Lemme 4.3 en 2002]

soit $\delta = \delta(x, z, w)$. De plus, soit $x^+ = x + \Delta x$ et $z^+ = z + \Delta z$. Alors

$$(x^+)^t z^+ = \|w\|^2 - \frac{\|q_v\|^2}{4},$$

donc

$$(x^+)^t z^+ \leq \|w\|^2.$$

Preuve. D'après

$$x^+ z^+ = w^2 - \frac{q_v^2}{4},$$

on obtient

$$(x^+)^t z^+ = e^t (x^+ z^+) = e^t w^2 - \frac{e^t q_v^2}{4} = \|w\|^2 - \frac{\|q_v\|^2}{4},$$

ce qui termine la preuve. \square

La mise à jour du paramètre w

Dans le lemme suivant, on discute l'influence de la mesure de proximité du processus de Newton suivi d'une étape le long du chemin pondéré. On suppose que chaque composante du vecteur w sera réduite d'un facteur constant $1 - \theta$.

Lemme 2.4. *[[6], Lemme 4.4 en 2002]*

Soit $\delta = \delta(x, z, w) < 1$ et $w^+ = (1 - \theta)w$, où $0 < \theta < 1$. Alors

$$\delta(x^+, z^+, w^+) \leq \frac{\theta}{1 - \theta} \sqrt{\sigma_c(w)n} + \frac{1}{1 - \theta} \delta(x^+, z^+, w),$$

de plus, si $\delta \leq \frac{1}{2}$, $\theta = \frac{1}{5\sqrt{\sigma_c(w)n}}$ et $n \geq 4$ alors on a

$$\delta(x^+, z^+, w^+) \leq \frac{1}{2}.$$

Preuve. On a

$$\begin{aligned} \delta(x^+, z^+, w^+) &= \frac{1}{\min(w^+)} \|w^+ - v^+\| \leq \frac{1}{\min(w^+)} \|w^+ - w\| + \frac{1}{\min(w^+)} \|w - v^+\|, \\ &= \frac{1}{(1 - \theta)\min(w)} \|\theta w\| + \frac{1}{1 - \theta} \delta(x^+, z^+, w), \\ &\leq \frac{\theta}{1 - \theta} \sqrt{\sigma_c(w)n} + \frac{1}{1 - \theta} \delta(x^+, z^+, w). \end{aligned}$$

Ainsi, la première partie du lemme est prouvée. Maintenant, on pose

$$\theta = \frac{1}{5\sqrt{\sigma_c(w)n}},$$

sachant que $\sigma_c(w) \geq 1$, et pour $n \geq 4$ on obtient $\theta \leq \frac{1}{10}$. De plus $\delta \leq \frac{1}{2}$ alors d'après le Lemme 2.2 on a

$$\delta(x^+, z^+, w) \leq \frac{1}{4}.$$

Enfin, les relations précédentes conduisent à

$$\delta(x^+, z^+, w^+) \leq \frac{1}{2}.$$

Ce qui termine la preuve. □

Proposition 2.2. *Pour chaque itération k , on a*

$$\sigma_c(w^k) = \sigma_c(w^0).$$

Preuve. À l'itération k , on a :

$$\begin{aligned} \sigma_c(w^k) &= \frac{\max((w^k)^2)}{\min((w^k)^2)}, \\ &= \frac{\max((1 - \theta)^{2k} (w^0)^2)}{\min((1 - \theta)^{2k} (w^0)^2)}, \\ &= \frac{(1 - \theta)^{2k} \max((w^0)^2)}{(1 - \theta)^{2k} \min((w^0)^2)}, \\ &= \sigma_c(w^0), \end{aligned}$$

pour toutes les itérations produites par l'Algorithme 2.4.2. Ce qui termine la preuve. Ainsi, un résultat immédiat du Lemme 2.4 est que pour $\theta = \frac{1}{5\sqrt{\sigma_c(w^0)n}}$, les conditions $(x, z) > 0$ sont maintenues tout au long de l'Algorithme 2.4.2. Par conséquent, l'Algorithme 2.4.2 est bien défini. \square

2.4.4 Analyse de la complexité

Le lemme suivant, présente une borne supérieure pour le nombre total d'itérations produitent par l'Algorithme 2.4.2.

Lemme 2.5. *[[6], Lemme 4.5 en 2002]*

On suppose que x^0 et z^0 sont strictement réalisable, et soit $w^0 = \sqrt{x^0 z^0}$. De plus, soit x^k et z^k les vecteurs obtenus après k ième itérations. Alors l'inégalité $(x^k)^t z^k \leq \epsilon$ est satisfaite si

$$k \geq \left\lceil \frac{1}{2\theta} \log \frac{(x^0)^t z^0}{\epsilon} \right\rceil.$$

Preuve. Après k itérations, on obtient :

$$w^k = (1 - \theta)^k w^0.$$

En utilisant le Lemme 2.3, on trouve

$$(x^k)^t z^k \leq \|w\|^2 = (1 - \theta)^{2k} \|w^0\|^2 = (1 - \theta)^{2k} (x^0)^t z^0,$$

donc $(x^k)^t z^k \leq \epsilon$ est vérifiée si

$$(1 - \theta)^{2k} (x^0)^t z^0 \leq \epsilon.$$

En prenant logarithme, on trouve

$$2k \log(1 - \theta) + \log((x^0)^t z^0) \leq \log \epsilon.$$

En utilisant l'inégalité $-\log(1 - \theta) \geq \theta$; $\forall 0 < \theta < 1$, on déduit que la relation ci-dessus est vérifiée si

$$2k\theta \geq \log((x^0)^t z^0) - \log \epsilon = \log \frac{(x^0)^t z^0}{\epsilon}.$$

Ce qui termine la preuve. \square

Pour préciser la valeur de θ spécifiée dans l'Algorithme 2.4.2 on présente le théorème suivante :

Théorème 2.4. *[[6], Darvay 2002]*

On suppose que le pair (x^0, z^0) est strictement réalisable, et soit $w^0 = \sqrt{x^0 z^0}$. Si $\theta = \frac{1}{5\sqrt{\sigma_c(w^0)n}}$ alors

l'Algorithme 2.4.2 nécessite au plus

$$\left\lceil \frac{5}{2} \sqrt{\sigma_c(w^0)n} \log \frac{(x^0)^t z^0}{\epsilon} \right\rceil,$$

itérations. Pour les vecteurs résultants, on a $x^t z \leq \epsilon$.

On suppose que le pair (x^0, z^0) est strictement réalisable, et soit $w^0 = \sqrt{x^0 z^0}$. Si $\theta = \frac{1}{5\sqrt{\sigma_c(w^0)}n}$ alors l'Algorithme 2.4.2 nécessite au plus

$$\left\lceil \frac{5}{2} \sqrt{\sigma_c(w^0)} n \log \frac{(x^0)^t z^0}{\epsilon} \right\rceil,$$

iterations. Pour les vecteurs résultants, on a $x^t z \leq \epsilon$.

Remarque 2.1. On remarque que le meilleur nombre d'itérations est obtenu si on suit la trajectoire centale c'est-à-dire :

$$\begin{aligned} \sigma_c(w^0) = 1 &= \frac{\max((w^0)^2)}{\min((w^0)^2)}, \\ &\Leftrightarrow \max((w^0)^2) = \min((w^0)^2), \\ &\Leftrightarrow \max(x^0 z^0) = \min(x^0 z^0), \end{aligned}$$

par conséquent, le nombre d'itérations est donné par :

$$O\left(\sqrt{n} \log \frac{(x^0)^t z^0}{\epsilon}\right).$$

Si le point initial n'est pas parfaitement centré, alors $\sigma_c(w^0) > 1$, ce qui fait la borne d'itérations est moins favorable.

3

Implémentation numérique

*D*ans ce chapitre, on s'intéresse aux expériences numériques en appliquant l'Algorithme 2.4.2 sur quelques problèmes linéaires pour voir la performance de l'algorithme proposé selon les valeurs de $\sigma_c(w^0)$.

Darvay et al [6] ont prouvé que la meilleure borne est obtenue si on suit la TC c'est à dire lorsque $\sigma_c(w^0) = 1$ et si $\sigma_c(w^0) > 1$ la borne d'itération est moins favorable donc on effectue les tests numériques pour vérifier l'efficacité de ces résultats théoriques.

Les expériences sont réalisées en utilisant le langage MATLAB exécuté sur la version (R2015a) sous Windows 10, avec un processeur Intel Core i5.

On note par la suite :

- $\delta(x, z, w)$: la mesure de proximité qui est associée à l'Algorithme 2.4.2.
- (x^0, y^0, z^0) : un point initial strictement réalisable et vérifié :

$$\delta(x^0, z^0, w^0) = \frac{\|w - v\|}{\min(w)} \leq \frac{1}{2}.$$

- $\sigma_c(w^0) = \frac{\max((w^0)^2)}{\min((w^0)^2)}$ tel que :

$$w^0 = \sqrt{x^0 z^0},$$

ou

$$w^0 = \xi e, \quad \text{avec } \xi > 0.$$

- (x^*, y^*, z^*) : la solution optimale du problème primal (P) et dual (D) respectivement.
- **iter** : Le nombre d'itérations générées par l'Algorithme 2.4.2.
- **CPU** : Le temps d'exécution de l'algorithme en secondes.

3.1 Tests numériques

Exemple 3.1. On considère le programme linéaire $n = 5$ et $m = 3$, et

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 42 \\ 41 \\ 24 \end{bmatrix},$$

$$c = [8 \ 8 \ 5 \ 2 \ 2]^t.$$

On prend comme un point réalisable initial

$$x^0 = [10 \ 9 \ 13 \ 7 \ 15]^t,$$

$$y^0 = [3 \ 1 \ 1]^t,$$

$$z^0 = [1 \ 2 \ 1 \ 2 \ 1]^t.$$

Une solution optimale approximative est donnée par :

$$x^* = [14.3333 \ 13.3333 \ 0.0001 \ 0 \ 10.6667]^t,$$

$$y^* = [3.3333 \ 1.3333 \ 2]^t,$$

$$z^* = [0 \ 0 \ 0.3333 \ 2 \ 0]^t.$$

La valeur optimale pour les deux problèmes est 242.6667.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-5}$.

w^0	$\sqrt{x^0 z^0}$	4 e
δ	0	0.2499
$\sigma_c(w^0)$	1,8	1
iter	127	95
CPU	0.0178	0.0164

TABLE 3.1: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.2. On considère le programme linéaire $n = 6$ et $m = 5$, et

$$A = \begin{bmatrix} 7 & 10 & 1 & 3 & -2 & 6 \\ -8 & -6 & 2 & 7 & 3 & 6 \\ 2 & 2 & -5 & 8 & -2 & 3 \\ 7 & 4 & 1 & -1 & 5 & -8 \\ -5 & 1 & -4 & 9 & 10 & 9 \end{bmatrix}, b = \begin{bmatrix} 122 \\ -12 \\ 17 \\ 27 \\ 60 \end{bmatrix},$$

$$c = [-2 \ 26 \ -11 \ 84 \ 77 \ 48]^t.$$

On prend comme un point réalisable initial

$$x^0 = [3 \ 7 \ 4 \ 1 \ 3 \ 5]^t,$$

$$y^0 = [2 \ 3 \ 1 \ 4 \ 5]^t,$$

$$z^0 = [3 \ 1 \ 2 \ 8 \ 4]^t.$$

Une solution optimale approximative est donnée par :

$$\begin{aligned} x^* &= \left[0 \quad 9.7728 \quad 4.2460 \quad 1.6842 \quad 2.2764 \quad 3.2544 \right]^t, \\ y^* &= \left[2.3417 \quad 3.9295 \quad 1.2587 \quad 4.6860 \quad 4.8982 \right]^t, \\ z^* &= \left[2.2157 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right]^t. \end{aligned}$$

La valeur optimale pour les deux problèmes est 680.3528.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-5}$.

w^0	$\sqrt{x^0 z^0}$	3 e
δ	0	0.2176
$\sigma_c(w^0)$	1,7143	1
iter	134	102
CPU	0.0218	0.0195

TABLE 3.2: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.3. On considère le programme linéaire $n = 7$, $m = 4$ et

$$\begin{aligned} A &= \begin{bmatrix} -8 & -2 & -8 & 6 & -3 & -1 & 7 \\ -5 & 10 & -2 & -9 & 4 & -4 & -5 \\ -8 & 1 & -1 & -3 & -8 & -6 & -6 \\ -4 & -3 & -4 & -2 & 6 & -3 & -1 \end{bmatrix}, b = \begin{bmatrix} -99 \\ -88 \\ -204 \\ -94 \end{bmatrix}, \\ c &= \left[-57 \quad -28 \quad -35 \quad 54 \quad -32 \quad -24 \quad 36 \right]^t. \end{aligned}$$

On prend comme un point réalisable initial

$$\begin{aligned} x^0 &= \left[9 \quad 6 \quad 8 \quad 6 \quad 5 \quad 7 \quad 5 \right]^t, \\ y^0 &= \left[5 \quad -3 \quad 4 \quad 2 \right]^t, \\ z^0 &= \left[8 \quad 14 \quad 11 \quad 13 \quad 15 \quad 9 \quad 12 \right]^t. \end{aligned}$$

Une solution optimale approximative est donnée par :

$$\begin{aligned} x^* &= \left[8.3137 \quad 2.9970 \quad 0 \quad 0 \quad 1.8489 \quad 20.9492 \quad 0 \right]^t, \\ y^* &= \left[5.6730 \quad -1.5396 \quad 1.8897 \quad 1.0491 \right]^t, \\ z^* &= \left[0 \quad 0 \quad 13.3912 \quad 13.8726 \quad 0 \quad 0 \quad 0.9779 \right]^t. \end{aligned}$$

La valeur optimale pour les deux problèmes est -910.2548.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-5}$.

w^0	$\sqrt{x^0 z^0}$	8 e
δ	0	0.2709
$\sigma_c(w^0)$	1,4667	1
iter	151	123
CPU	0.5538	0.0251

TABLE 3.3: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.4. On considère le programme linéaire $n = 9$, $m = 6$, et

$$A = \begin{bmatrix} -8 & 7 & -6 & 8 & -2 & 9 & 10 & 7 & -8 \\ -2 & -5 & 8 & 7 & 4 & 3 & -5 & 2 & 1 \\ 4 & -4 & 7 & 1 & 9 & 8 & 3 & 2 & 5 \\ 7 & 4 & -1 & 1 & 5 & -6 & -8 & 2 & 3 \\ -9 & 5 & 4 & 2 & 7 & -8 & -3 & -4 & 5 \\ 1 & 8 & -5 & -3 & 1 & 2 & 7 & 9 & -3 \end{bmatrix}, b = \begin{bmatrix} -91 \\ 67 \\ 240 \\ 126 \\ 17 \\ 51 \end{bmatrix},$$

$$c = \begin{bmatrix} -68 & 78 & 25 & 127 & 86 & 59 & 16 & 88 & -17 \end{bmatrix}^t.$$

On prend comme un point réalisable initial

$$x^0 = \begin{bmatrix} 10 & 1 & 4 & 2 & 11 & 1 & 4 & 5 & 9 \end{bmatrix}^t,$$

$$y^0 = \begin{bmatrix} 8 & 7 & 2 & 5 & 4 & 2 \end{bmatrix}^t,$$

$$z^0 = \begin{bmatrix} 1 & 9 & 2 & 5 & 1 & 8 & 3 & 2 & 1 \end{bmatrix}^t.$$

Une solution optimale approximative est donnée par :

$$x^* = \begin{bmatrix} 10.6395 & 0 & 4.5250 & 0 & 16.5498 & 0.1372 & 3.7816 & 2.1878 & 0 \end{bmatrix}^t,$$

$$y^* = \begin{bmatrix} 8.2826 & 7.1190 & 2.3845 & 4.9568 & 3.7238 & 1.7774 \end{bmatrix}^t,$$

$$z^* = \begin{bmatrix} 0 & 12.4897 & 0 & 1.4491 & 0 & 0.0001 & 0 & 0 & 2.0616 \end{bmatrix}^t.$$

La valeur optimale pour les deux problèmes est $1.0741 \cdot 10^3$.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-4}$.

w^0	$\sqrt{x^0 z^0}$	3.2e
δ	0	0.2082
$\sigma_c(w^0)$	1,5	1
iter	137	112
CPU	0.0622	0.0264

TABLE 3.4: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.5. On considère le programme linéaire $n = 10$, $m = 8$, et

$$A = \begin{bmatrix} 3 & -5 & 2 & -8 & -5 & 2 & 6 & 10 & 5 & 2 \\ 2 & -8 & -6 & -7 & 6 & 5 & 5 & 6 & -9 & -8 \\ 6 & 1 & 9 & 5 & -2 & 3 & 10 & -5 & 6 & -7 \\ 6 & 3 & -2 & 1 & -4 & 7 & -8 & 10 & 8 & 4 \\ 1 & -3 & -1 & 5 & -8 & -4 & 4 & 1 & 4 & -2 \\ 4 & -1 & 3 & -2 & 5 & -8 & 4 & 1 & -6 & 0 \\ 4 & -2 & 6 & -8 & 8 & 8 & -2 & -1 & 6 & 7 \\ 8 & 0 & -9 & -5 & 5 & 4 & -8 & -6 & 5 & 5 \end{bmatrix}, b = \begin{bmatrix} 47 \\ 79 \\ 63 \\ 107 \\ -108 \\ 85 \\ 246 \\ 155 \end{bmatrix},$$

$$c = [223 \quad -115 \quad 12 \quad -210 \quad 106 \quad 204 \quad 68 \quad 125 \quad 117 \quad 21]^t.$$

On prend comme un point réalisable initial

$$x^0 = [12 \quad 1 \quad 2 \quad 1 \quad 14 \quad 5 \quad 3 \quad 4 \quad 2 \quad 6]^t,$$

$$y^0 = [9 \quad 10 \quad 6 \quad 5 \quad 1 \quad 3 \quad 10 \quad 7]^t,$$

$$z^0 = [1 \quad 15 \quad 5 \quad 13 \quad 1 \quad 3 \quad 4 \quad 3 \quad 5 \quad 2]^t.$$

Une solution optimale approximative est donnée par :

$$x^* = [12.8335 \quad 0 \quad 3.4809 \quad 0 \quad 12.2422 \quad 5.0197 \quad 1.0532 \quad 3.1823 \quad 0.8708 \quad 5.1070]^t,$$

$$y^* = [8.3769 \quad 10.5129 \quad 5.6116 \quad 5.0390 \quad 2.9947 \quad 3.0126 \quad 11.6177 \quad 6.4280]^t,$$

$$z^* = [0 \quad 25.4914 \quad 0 \quad 13.6418 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^t,$$

La valeur optimale pour les deux problèmes est $5.9039 \cdot 10^3$.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-4}$.

w^0	$\sqrt{x^0 z^0}$	3.5e
δ	0	0.2178
$\sigma_c(w^0)$	1,5	1
iter	148	120
CPU	0.0333	0.0231

TABLE 3.5: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.6.

On considère le programme linéaire $n = 15$, $m = 13$ et

$$A = \begin{bmatrix} 10 & -5 & 9 & -5 & -14 & -2 & 13 & 11 & 15 & 12 & 25 & 16 & 32 & -17 & -22 \\ -2 & 8 & 5 & 7 & 16 & 25 & -15 & 11 & -9 & 8 & 32 & 18 & -15 & 51 & -10 \\ 1 & 9 & 6 & 3 & -5 & 9 & -7 & -12 & 16 & -27 & 36 & -24 & -29 & 38 & 42 \\ 9 & 8 & -12 & -31 & 4 & -7 & 8 & 20 & 48 & -64 & 18 & 19 & -27 & 48 & 4 \\ -1 & -2 & -5 & 8 & 4 & -4 & -2 & 34 & -24 & -25 & 17 & 29 & -31 & 26 & 56 \\ 8 & 10 & -9 & -5 & 10 & 4 & -8 & -6 & 6 & 7 & -1 & 3 & -2 & 5 & -8 \\ 4 & 1 & -6 & 4 & -2 & 6 & -8 & 8 & 8 & 12 & -2 & -1 & 6 & 17 & 7 \\ 8 & 11 & 10 & -9 & -5 & 7 & 4 & -8 & -6 & 5 & 26 & 13 & -9 & -36 & 18 \\ 47 & 79 & 63 & 10 & -18 & 53 & 24 & 15 & 68 & 21 & 14 & 13 & -15 & 17 & -7 \\ 22 & 14 & 17 & -29 & -30 & 21 & 45 & 36 & 27 & -31 & -24 & 12 & 9 & 19 & 22 \\ -2 & -5 & -2 & 11 & 12 & 16 & 32 & -17 & 18 & 5 & -6 & 2 & 21 & 13 & -18 \\ 8 & 19 & -12 & -31 & 10 & 7 & 3 & 1 & -5 & 14 & 12 & 6 & -11 & 18 & 11 \\ 14 & 51 & 23 & -19 & 11 & 6 & 34 & -18 & 13 & 19 & 10 & 17 & 29 & 26 & 9 \end{bmatrix}$$

$$b = [317 \ 1320 \ 835 \ 732 \ 1847 \ 45 \ 441 \ 432 \ 1971 \ 888 \ -104 \ 838 \ 1279]^t,$$

$$c = [1351 \ 1897 \ 1177 \ -741 \ -363 \ 1853 \ 1970 \ 668 \ 2181 \ -114 \ 792 \ 995 \ 343 \ 1968 \ 321]^t.$$

On prend comme un point réalisable initial $x^0 = [13 \ 1 \ 8 \ 1 \ 15 \ 2 \ 5 \ 17 \ 1 \ 9 \ 13 \ 7 \ 1 \ 9 \ 16]^t$,

$$y^0 = [9 \ 10 \ 6 \ 5 \ 1 \ 3 \ 10 \ 7 \ 12 \ 19 \ 21 \ 9 \ 7]^t, \quad z^0 = [1 \ 16 \ 2 \ 13 \ 1 \ 9 \ 3 \ 1 \ 15 \ 2 \ 1 \ 2 \ 18 \ 2 \ 1]^t,$$

Une solution optimale approximative est donnée par :

$$x^* = [14.0125 \ 0 \ 8.2989 \ 0.9732 \ 13.3889 \ 1.1786 \ 5.1504 \ 15.4611 \ 0.8829 \ 9.6936 \ 12.5683 \ 8.2176 \ 0 \ 9.8387 \ 15.826]^t$$

$$y^* = [8.8205 \ 9.9097 \ 5.876 \ 5.17 \ 1.05 \ 2.5128 \ 10.5706 \ 7.2279 \ 12.1843 \ 18.7913 \ 21.3726 \ 8.9889 \ 6.8437]^t,$$

$$z^* = [0 \ 15.8769 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 23.8163 \ 0 \ 0]^t,$$

La valeur optimale pour les deux problèmes est $8.9169.10^4$.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-4}$.

w^0	$\sqrt{x^0 z^0}$	3.9e
δ	0	0.2345
$\sigma_c(w^0)$	1,3846	1
iter	181	154
CPU	0.0294	0.0237

TABLE 3.6: Résultat comparatif basé sur le choix de w^0 .

Exemple 3.7.

On considère le programme linéaire $n = 18, m = 17$ et

$$A = \begin{bmatrix} -2 & -5 & 11 & 16 & 14 & 10 & -9 & -4 & 12 & 15 & 17 & 42 & 36 & -15 & 4 & 21 & 21 & 33 \\ 51 & -10 & 18 & 32 & 19 & -13 & 16 & 12 & 10 & 2 & 8 & 5 & 15 & 7 & 35 & -15 & -19 & 14 \\ 56 & 26 & -32 & 17 & -25 & -6 & 8 & -4 & 16 & 27 & -39 & 45 & 10 & 9 & 6 & -12 & 7 & 9 \\ 19 & 11 & 16 & 12 & -5 & -17 & 14 & -11 & -19 & 13 & 21 & 8 & -12 & 4 & -8 & 7 & 23 & -18 \\ 20 & 48 & -63 & 14 & 19 & 36 & -27 & 4 & 41 & 27 & 50 & 63 & -20 & 32 & 66 & 45 & 34 & 19 \\ 14 & 22 & 17 & -29 & 2 & -17 & -1 & 2 & 5 & 18 & 34 & 13 & 21 & 9 & 14 & 34 & 13 & -27 \\ 10 & 5 & 1 & 17 & 19 & 25 & 22 & -21 & -30 & -24 & -25 & 7 & 9 & -3 & 6 & 5 & -8 & -6 \\ 46 & 70 & 9 & -59 & 17 & 81 & 22 & 6 & 10 & 5 & -9 & 13 & 21 & -11 & 33 & -56 & 14 & 16 \\ 15 & -59 & -23 & 31 & 18 & 24 & 67 & 13 & 14 & 54 & -39 & 10 & 4 & -6 & -21 & 37 & 30 & 17 \\ -44 & 21 & -88 & 91 & -33 & -29 & 13 & 21 & 20 & 38 & -8 & -36 & -18 & 13 & 27 & 45 & 12 & 23 \\ 51 & 23 & 14 & -68 & 55 & -28 & 26 & -11 & 6 & -7 & 45 & 17 & -35 & -9 & 22 & -14 & 10 & 9 \\ 11 & -16 & 23 & -19 & -29 & 25 & 9 & 18 & 21 & 49 & 14 & -13 & 6 & -3 & 18 & 41 & 48 & 53 \\ 26 & 17 & -10 & -1 & 11 & 20 & 12 & 14 & -16 & 6 & 34 & 11 & -6 & 4 & 22 & 60 & 46 & 51 \\ 17 & 6 & -17 & -28 & 6 & 18 & 52 & -12 & -21 & 16 & 77 & 29 & 26 & 50 & -30 & 11 & 16 & -2 \\ 8 & 19 & 61 & 80 & 71 & 26 & -41 & -9 & -5 & 27 & 14 & -88 & -51 & 15 & 20 & 31 & 22 & -17 \\ 18 & -29 & 17 & 31 & 6 & -34 & 10 & -17 & 26 & 9 & -7 & 33 & -22 & 37 & 41 & 10 & -11 & 22 \\ 14 & -26 & 18 & 11 & -2 & 5 & -4 & 12 & 16 & 35 & -19 & 12 & -14 & 9 & 26 & 17 & 14 & -19 \end{bmatrix}$$

$$b = [1652 \ 1591 \ 980 \ 386 \ 3060 \ 1029 \ 78 \ -16 \ 1236 \ 632 \ 630 \ 1027 \ 1749 \ 666 \ 1062 \ 1679 \ 992]^t,$$

$$c = [2063 \ 1965 \ 751 \ 1138 \ 1942 \ -21 \ -211 \ -475 \ 1682 \ 1569 \ 3256 \ 1326 \ -757 \ 1294 \ 3740 \ 1293 \ 1472 \ 2029]^t.$$

On prend comme un point réalisable initial

$$x^0 = [11 \ 1 \ 5 \ 12 \ 7 \ 1 \ 2 \ 5 \ 12 \ 1 \ 7 \ 13 \ 4 \ 1 \ 7 \ 14 \ 1 \ 2]^t,$$

$$y^0 = [22 \ 15 \ -5 \ 12 \ 17 \ 3 \ 4 \ 8 \ -12 \ 11 \ 17 \ 16 \ -8 \ 4 \ 11 \ 20 \ -6]^t,$$

$$z^0 = [1 \ 14 \ 2 \ 1 \ 2 \ 14 \ 7 \ 2 \ 1 \ 13 \ 2 \ 1 \ 3 \ 12 \ 2 \ 1 \ 13 \ 7]^t.$$

Une solution optimale approximative est donnée par :

$$x^* = [10.9407 \ 1.3032 \ 5.2473 \ 12.0278 \ 6.8655 \ 1.22 \ 2.1124 \ 5.1849 \ 11.8104 \ 1.4931 \ 7.1376 \ 13.2997 \ 3.5676 \ 0.6783 \ 6.7844 \ 14.3127 \ 0 \ 2.1714]^t$$

$$y^* = [22.1341 \ 14.7693 \ -4.9554 \ 11.9192 \ 16.7897 \ 3.0414 \ 3.7455 \ 8.3139 \ -12.0118 \ 11.1156 \ 16.8928 \ 15.7754 \ -7.7422 \ 4.1773 \ 11.1 \ 20.2289 \ -5.8313]^t,$$

$$z^* = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1.6056 \ 0]^t,$$

La valeur optimale pour les deux problèmes est $1.6270.10^5$.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.4.2 pour les différentes valeurs de w^0 avec $\epsilon = 10^{-4}$.

w^0	$\sqrt{x^0 z^0}$	3.6e
δ	0	0.2321
$\sigma_c(w^0)$	1,4	1
iter	200	169
CPU	0.0848	0.0463

TABLE 3.7: Résultat comparatif basé sur le choix de w^0 .

Conclusion générale

Dans ce travail, on s'intéresse à résoudre un problème linéaire (PL) par une méthode primale-duale de trajectoire centrale avec poids basée sur une nouvelle direction de recherche qui est présenté par Darvay en 2002 [6]. Dont le but est de généraliser la méthode de trajectoire centrale classique qui est déjà présenté par Roos 1997 [19] et d'avoir nouvelles directions de recherche, Darvay a été proposé une nouvelle transformation algébrique $\psi(xz) = \psi(w^2)$ où $\psi(t) = \sqrt{t}$ au lieu d'utiliser l'équation de centralité $xz = \mu\epsilon$, avec $xz = w^2$.

Il a montré que cet algorithme a une meilleure complexité polynomiale qui est de l'ordre

$$O \left[\frac{5}{2} \sqrt{\sigma_c(w^0)} n \log \frac{(x^0)^t z^0}{\epsilon} \right].$$

Dont paramètres qui ont un rôle important dans l'algorithme :

$$w^0 = \sqrt{x^0 z^0} \quad \text{et} \quad \beta = \frac{1}{2} \quad \text{avec} \quad \theta = \frac{1}{5\sqrt{\sigma_c(w^0)} n}.$$

La difficulté de cette méthode est justifié comment trouvé le point initial qui vérifié :

$$(x^0, y^0, z^0): \begin{cases} x^0 > 0, z^0 > 0, \\ Ax^0 = b, A^t y^0 + z^0 = c, \\ \delta(x^0, y^0, z^0) < \frac{1}{2}. \end{cases}$$

Notons que notre étude à réalisé les contributions suivants :

- On a appliqué cet algorithme sur quelque problèmes linéaires pour avoir son efficacité.
- On compare entre le choix théorique de $\sigma_c(w^0)$ pour voir l'influence de $\sigma_c(w^0)$ sur le comportement numérique de cet algorithme.

Bibliographie

- [1] M. Achache, *A new primal-dual path-following method for convex quadratic programming*, *Comput. Appl. Math.* 25(1) (2006) 97–110.
- [2] N. Andrei. *Mathematical Programming. Interior Point Methods*. Editura Tehnica, Bucuresti, 1999. (In Romanian).
- [3] Zs. Darvay. *A new algorithm for solving self-dual linear optimization problems*. *Studia Universitatis Babe_s-Bolyai, Series Informatica*, 47(1) :15–26, 2002.
- [4] Zs. Darvay. *A new class of search directions for linear optimization*. In *Proceedings of Abstracts, McMaster Optimizations Conference : Theory and Applications held at McMaster University Hamilton, Ontario, Canada, page 18, August 1-3, 2002*. Submitted to *European Journal of Operational Research*.
- [5] Zs. Darvay. *A new interior point algorithms in linear programming*. *Adv. Model. Optim*, 2003, vol. 5, no 1, p. 51-92.
- [6] Zs. Darvay. *A weighted path following method for linear optimization*. *Studia Universitatis Babes-Bolyai, Informatica*, volum XLVIL Number 2, 2002
- [7] Zs. Darvay. *Interior Point Methods in Linear Programming*. ELTE, Budapest, 1997. (In Hungarian).
- [8] J. Ding and T.Y. Li. *An algorithm based on weighted logarithmic barrier functions for linear complementarity problems*. *Arabian Journal for Science and Engineering*, 15(4) :679–685, 1990.
- [9] P. HUARD, *Programmation mathématique convexe*. *Revue française d’informatique et de recherche opérationnelle*, 1968, vol. 2, no 7, p. 43-59.
- [10] B. Jansen. *Interior Point Techniques in Optimization. Complexity, Sensitivity and Algorithms*. Kluwer Academic Publishers, 1997.
- [11] B. Jansen, C. Roos, T. Terlaky, and J.-Ph. Vial. *Primal-dual target-following algorithms for linear programming*. *Annals of Operations Research*, 62 :197–231, 1996.
- [12] N.K. Karmarkar. *A new polinomial-time algorithm for linear programming*. *Combinatorica*, 4 :373–395, 1984.
- [13] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, volume 538 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 1991.
- [14] Z. Ramdani, *Cours d’optimisation non linéaire*, Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj(2021).
- [15] Z. Ramdani, *Cours de programmation linéaire*, Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj (2021).
- [16] C. ROOS, *A full-Newton step $O(n)$ infeasible interior-point algorithm for linear optimization*. *SIAM Journal on Optimization*, 2006, vol. 16, no 4, p. 1110-1136.

- [17] C. ROOS, *An Improved and Simplified Full-Newton Step $O(n)$ Infeasible Interior-Point Method for Linear Optimization*. *SIAM Journal on Optimization*, 2015, vol. 25, no 1, p. 102-114.
- [18] C. Roos, and D. den Hertog. *A polynomial method of approximate weighted centers fo linear programming*. *Technical Report 89-13, Faculty of Technical Mathematics and Informatics, TU Delft, NL-2628 BL Delft, The Netherlands, 1994*.
- [19] C. ROOS, TERLAKY, Tamás, et VIAL, J.-Ph. *Interior point methods for linear optimization*. 2005.
- [20] C. Roos, T. Terlaky, and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization. An Interior Approach*. John Wiley Sons, Chichester, UK, 1997.
- [21] Gy. Sonnevend. *An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*. In A. Prékopa, J. Szelezsan, and B. Strazicky, editors, *System Modelling and Optimization : Proceedings of the 12th IFIP-Conference held in Budapest, Hungay, September 1985, volume 84 of Lecture Notes in Control and Information Sciences, pages 866–876*. Springer Verlag, Berlin, West-Germany, 1986.
- [22] Y. Ye. *Interior Point Algorithms, Theory and Analysis*. John Wiley Sons, Chichester, UK, 1997.
- [23] S.J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, USA, 1997.

Résumé

Dans ce travail, on a étudié un algorithme réalisable primal-dual de trajectoire central avec poids basé sur une nouvelle direction de recherche pour résoudre un problème linéaire [voir Darvay6]. L'idée principale de cet algorithme est de remplacer l'équation de centralité $xz = \mu e$ par le poids $xz = w^2$ ou $w > 0$ et introduire une transformation algébrique à cette dernière équation dont le but est de généraliser la méthode de trajectoire centrale classique qui est déjà présenté par Roos 1997 [19] et d'avoir nouvelles directions de recherche.

A chaque itération, on utilise le pas de Newton complet pour trouver une solution approximative. cette étude est suivie par des tests numériques comparative selon le choix de w^0 pour montré l'efficacité de l'algorithme étudié.

Mots clés : Méthode de point intérieur, Programmation linéaire, Algorithme réalisable, Méthode de trajectoire centrale, Nouvelle direction de recherche.

Abstract

In this work, we have studied a feasible primal-dual central path algorithm with weights based on new search direction to solve a linear problem (LP) (see [6]). The basic idea of this algorithm is to replace the centrality equation $xz = \mu e$ with the weight equation $xz = w^2$ where $w > 0$, and introduce an algebraic transformation to this latter equation and introduce an algebraic transformation to this latter equation aimed at generalizing the classical path-following method already presented by Roos 1997 [19] and obtaining new search directions. At each iteration, we use the full Newton step to find an approximate solution. This study is followed by comparative numerical tests based on the choice of w^0 to demonstrate the efficiency of the studied algorithm.

• **keywords :** Interior point method, Linear programming, Feasible algorithm, Central path method, New research direction.

المخلص

في هذا العمل قمنا بدراسة خوارزمية قابلة للتنفيذ للمسار المركزي بثقل تركز على اتجاهات بحث جديدة لحل مشكلة البرمجة الخطية. تركز الفكرة الأساسية لهذه الخوارزمية على استبدال المعادلة المركزية $xz = \mu e$ بمعادلة الوزن $xz = w^2$ حيث $w > 0$ وإدخال تحويل جبري على هذه المعادلة الأخيرة بهدف تعميم طريقة المسار المركزي الكلاسيكية والحصول على اتجاهات بحث جديدة.

في كل تكرار نستخدم خطوة نيوتن الكاملة لإيجاد حل تقريبي. هذه الدراسة متبوعة باختبارات عددية لاثبات فعالية هذه الخوارزمية. الكلمات المفتاحية: طريقة النقاط الداخلية, برمجة خطية, خوارزمية قابلة للتنفيذ, اتجاه بحثي جديد

