REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

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

THESE

Présentée par :

Menniche Linda

Pour obtenir le diplôme de Doctorat en Sciences

OPTION

MATHEMATIQUES APPLIQUEES

THEME

Etude théorique et numérique d'une classe de méthodes de points intérieurs pour la programmation linéaire

Soutenue Le : 02/02/2017
devant le jury composé de :

Président	**: Mr. N. BENSALEM**	**Prof. U.F.A. Sétif-1.**
Rapporteur	**: Mr. D. BENTERKI**	**Prof. U.F.A. Sétif-1.**
Examinateurs	**: Mme. Y. LASKRI**	**Prof. E.S.T.I. Annaba.**
	Mr. B. BOUDERAH	**Prof. U.M.B. M'sila.**

Année 2016/ 2017

REMERCIMENTS

J'exprime ma profonde reconnaissance à Monsieur **DJ. BENTERKI**, professeur en mathématique appliquées à l'université de Ferhat Abbes Sétif-1, qui a dirigé ce travail, pour son aide continue, sa disponibilité et ses conseils qui ont contribué à l'élaboration de cette thèse.

Je remercie Monsieur **N. BENSALEM** professeur à l'université de Ferhat Abbes-1, qui m'honore en acceptant d'assurer la présidence du jury, malgré ses obligations professionnelles. Qu'il trouve ici l'expression de ma respectueuse reconnaissance.

Mes vifs remerciements vont également à Madame **Y. LASKRI** professeur à l'université de Annaba et monsieur **B. BOUDERAH** professeur à l'université de M'sila, qui ont accepté de juger ce travail et d'êtres les membres du jury. Qu'ils veuillent bien agréer l'expression de mes sincères et respectueuses considérations.

Je remercie aussi tous les membres de l'équipe de l'optimisation, pour leur disponibilité, en particulier Monsieur **B.MERIKHI**, qui ma apporté son aide, avec beaucoup de gentillesse et de sympathie.

Table des matières

Introduction **3**

1 Préliminaires et notions fondamentales **7**

1.1 Préliminaires 7

1.1.1 Ensembles et fonctions affines 7

1.1.2 Notions de convexité 8

1.1.3 Convexité et dérivée 10

1.1.4 Fonction barrière 13

1.1.5 Programmation mathématique 13

1.2 Programmation linéaire 17

1.3 Formes usuelles d'un programme linéaire 19

1.3.1 Forme standard 19

1.3.2 Forme canonique 19

1.4 Dualité en programmation linéaire 20

2 Méthodes de résolution d'un programme linéaire **24**

2.1 Domaines d'applications de la programmation linéaire 24

2.1.1 Exemples concrets de problèmes de programmation linéaire 26

2.2 Méthode de simplexe 34

2.2.1 Caractéristiques de l'algorithme du simplexe 34

2.3 Méthodes de points intérieurs 35

2.3.1 Méthodes affines 36

2.3.2 Algorithme de la méthode affine primale 39
2.3.3 Méthode de réduction du potentiel 42
2.3.4 Méthodes de trajectoire centrale 43
2.3.5 Méthodes de trajectoire centrale primales-duales 45
2.4 Résultats de convergence 51

3 Méthode barrière logarithmique via les fonctions majorantes 52

3.1 Introduction 52
3.2 Aspect théorique du problème (DL_r) 54
3.2.1 Existence de la solution du problème (DL_r) 54
3.2.2 Unicité de la solution du problème (DL_r) 55
3.2.3 Convergence de (DL_r) vers (DL) lorsque r tend vers zéro 56
3.3 Aspect numérique du problème (DL_r) 57
3.3.1 Calcul du pas de déplacement t_k 59
3.3.2 Calcul des différentes valeurs de $\widehat{t}$ 61
3.4 Fonctions majorantes de θ 62
3.4.1 Première fonction majorante 63
3.4.2 Seconde fonction majorante 64
3.4.3 Troisième fonction majorante 65
3.5 Tests numériques 68
3.5.1 Exemples à dimension fixe 69
3.5.2 Exemple à dimension variable 72

Conclusion 73

Bibliographie 75

Introduction

La programmation linéaire est certainement l'un des plus beaux succès de la recherche opérationnelle. Il provient d'une part, de la puissance de modélisation qu'elle offre malgré la limite inhérente qu'impose la linéarité des fonctions impliquées, et d'autre part, de la richesse de la théorie qu'elle a initié et qui a permet le développement d'algorithme extrêmement efficace pour sa résolution. Depuis sa formulation et le développement de la méthode du simplexe pour sa résolution vers la fin des années 40, la programmation linéaire demeure le modèle d'optimisation le plus utilisé par les décideurs, ce qui est certainement dû à la robustesse et la stabilité des algorithmes disponibles. La nature combinatoire de l'algorithme du simplexe a probablement fait en sorte que la programmation linéaire a été longtemps considérée comme une classe de problème distincte de la programmation non linéaire. Lorsque Klee et Minty (1972) ont trouvé un exemple qui montrait pour la première fois que la méthode du simplexe pouvait prendre un nombre exponentiel d'itérations (une question demeure toute fois ouverte : existe t-il un algorithme de type pivot et de complexité polynomiale pour résoudre convenablement un programme linéaire ?). A cet égard, la recherche d'un algorithme polynomial pour la programmation linéaire était lancée. La réponse est venue de Khachiyan en (1979) lorsqu'il a montré que la méthode des ellipsoïdes (un algorithme de points intérieurs développé au début des années 70 pour la programmation non linéaire) était de complexité globale O(n^4 L) lorsqu'elle est appliquée à la programmation linéaire, où L est le nombre de bits requis

pour stocker (et traiter) les données. Cependant, et tandis que la méthode du simplexe de complexité exponentielle dans le pire des cas, fonctionnait bien en pratique.

En (1984), Karmarkar a proposé un algorithme polynomial de complexité ($n^{3.5}$ L) efficace en pratique, basé sur une méthode de points intérieurs. L'intérêt pour ces méthodes principalement développées depuis les années 60 voir Dikin (1967) et Fiacco et McCormik (1968)[13], a connu un renouveau pour les problèmes non linéaires et a ouvert un nouveau domaine pour les problèmes linéaires.

Dès lors, les problèmes de programmation linéaire et non linéaire sont devenus des thèmes de recherche très convoités depuis la relance des méthodes de points intérieurs issues des nouvelles investigations apportées par Karmarkar et autres. En effet, ces méthodes avec leurs éléments nouveaux s'avèrent très intéressantes pour un traitement descend de la programmation linéaire allant de l'aspect théorique pur à l'aspect numérique proprement dit. Les méthodes de points intérieurs s'avèrent efficaces pour les problèmes d'optimisation à grande taille à cause de leur caractère polynomial. Den Herty (1994) a classé les méthodes points intérieurs en trois catégories : méthodes affines, méthodes de réduction de potentiel et les méthodes de trajectoire centrale [32]. Les méthodes de points intérieurs ont fait l'objet de plusieurs travaux de recherche, ceux effectués par Den Hertog [11], Nestrov et Nemirovski [26], Roos, Terlaky et Vial [29], Wright [32], Ye [33],...etc. Plusieurs algorithmes ont été proposé pour résoudre les problèmes de programmation linéaire, par les méthodes projectives et leurs variantes [7, 18, 25, 34], les méthodes de trajectoire centrale [1, 4, 5, 10, 16] et les méthodes barrières logarithmiques [8, 21, 23, 24]. Notre travail est basé sur ce dernier type de méthodes de points intérieurs. Le principal obstacle pour construire une itération est la détermination et le calcul du pas de déplacement. Plusieurs alternatives sont proposées pour résoudre ce problème parmi lesquelles les méthodes de recherche linéaire [19], malheureusement le calcul du pas de déplacement par ces méthodes est coûteux et encore plus délicat dans le cas des problèmes de programmation semi-définie linéaire. Nous proposons dans ce travail une approche barrière logarithmique pour

laquelle, on introduit une procédure originale pour le calcul du pas de déplacement basée sur les fonctions majorantes, et ceci en s'inspirant des travaux de J.P. Crouzeix et B. Merikhi en programmation semi-définie linéaire [6, 8].

Cette thèse est composée de trois chapitres organisés comme suit :

Le premier chapitre contient une présentation générale de quelques notions de base nécessaires par la suite telle que l'analyse convexe, la programmation mathématique, les conditions d'optimalités et la programmation linéaire.

Le deuxième chapitre est réservé pour les domaines d'application de la programmation linéaire et les méthodes numériques de résolution. Nous mettons l'accent sur les différentes méthodes de points intérieurs pour la programmation linéaire. En particulier, nous présentons d'une manière détaillée le principe de ces méthodes, les algorithmes appropriés et les résultats de convergence. Nous dégageons également les inconvénients et les difficultés rencontrés tant sur le plan numérique que théorique.

Le troisième chapitre contient le travail essentiel de cette thèse. On s'intéresse à la résolution d'un problème de programmation linéaire par les méthodes de point intérieurs de type barrière logarithmique. Le principe de ces méthodes étant de perturbé le problème de programmation linéaire initial (DL) par un autre problème de programmation linéaire (DL_r), moyennant une fonction barrière. On montre par la suite que le problème perturbé (DL_r) possède une solution optimale pour chaque paramètre barrière r, et que la suite de solution optimale en r tend vers une solution optimale du problème initial (DL) lorsque r tend vers 0. La deuxième partie de ce chapitre est réservée à la résolution numérique du problème (DL) par une méthode de points intérieurs de type barrière logarithmique. Rappelons que l'inconvénient majeur de ce type de méthodes est le calcul effectif et efficace du pas de déplacement. Un tel pas joue un rôle important dans le comportement de l'algorithme (vitesse de convergence, coût des itérations). Les méthodes les plus populaires pour le calcul du pas de déplacement sont les méthodes de recherche linéaire, malheureusement, ces méthodes sont coûteuses en volume calculatoire voir même impossible dans

certains problèmes de programmation semi-définie. Le principe général des méthodes de recherches linéaires est la minimisation de la fonction unidimensionnelle barrière logarithmique par des procédures techniques connues telle que, Armijou, Goldestein Price, Fletcher,...etc. Pour remédier aux difficultés des recherches linéaires, nous proposons une fonction approximante de la fonction barrière dont le but de calculer aisément le pas de déplacement. Dans ce sens, nous avons proposé trois fonctions approximantes majorantes et nous avons donné la forme explicite des pas de déplacement pour chaque fonction sans faire recours aux méthodes de recherche linéaire. Enfin, nous avons présenté à la fin de ce chapitre des tests numériques d'ordre comparatif entre les trois fonctions majorantes afin de confirmer les propos théoriques.

Chapitre 1

Préliminaires et notions fondamentales

Dans ce chapitre, nous rappelons quelques notions de base sur l'analyse convexe et la programmation linéaire. Nous rappelons également, les résultats de dualité en programmation linéaire.

1.1 Préliminaires

1.1.1 Ensembles et fonctions affines

Définition 1.1.1. ***(Ensemble affine)***
Un sous-ensemble D de $\mathbb{R}^n$ est dit affine si :

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

Les ensembles affines élémentaires sont : ϕ, $\mathbb{R}^n$, $\{x\}$ avec $(x \in \mathbb{R}^n)$ et chaque sous-espace vectoriel de $\mathbb{R}^n$.

Définition 1.1.2. ***(Combinaison affine)***
Soient $x_1, ..., x_m$, des points quelconques de $\mathbb{R}^n$, on dit que $x \in \mathbb{R}^n$ est une combinaison affine de $x_1, ..., x_m$, s'il existe $(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbb{R}^m$ tels que

$$x = \sum_{i=1}^{m} \lambda_i x_i \text{ avec } \sum_{i=1}^{m} \lambda_i = 1.$$

Proposition 1.1.1. *Un sous-ensemble $D \in \mathbb{R}^n$ est dit affine si et seulement s'il contient toutes les combinaisons affines de ses éléments.*

Définition 1.1.3. ***(Enveloppe affine)***
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 $aff(S)$. Autrement dit $aff(S)$ est la plus petite partie affine de $\mathbb{R}^n$ contenant S.

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

Nous avons par définition $dim(S) = dim(aff(S))$.
Si $S \neq \phi$, alors $aff(S) \neq \phi$ (puisque $S \subseteq aff(S)$).

Un sous-ensemble S de $\mathbb{R}^n$ est affine si et seulement si $S = aff(S)$.

Théorème 1.1.2. *Soit S une partie non vide de $\mathbb{R}^n$, alors*

$$aff(S) = \{x \in \mathbb{R}^n / x = \sum_{i=1}^{m} \lambda_i x_i, \;\; x_i \in S, \;\; \sum_{i=1}^{m} \lambda_i = 1\}.$$

Définition 1.1.4. ***(Fonctions affines)***
Une fonction f définie de $\mathbb{R}^n$ dans $\mathbb{R}^m$ est dite affine si

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

Théorème 1.1.3. *Toute fonction affine f de $\mathbb{R}^n$ dans $\mathbb{R}^m$ est de la forme $f(x) = Ax + b$, où $A \in$ à l'ensemble des matrices réelle de taille $(m \times n)$ $\mathcal{M}_{m,n}$ et $b \in \mathbb{R}^m$.*

Proposition 1.1.4. *Une fonction affine f de $\mathbb{R}^n$ dans $\mathbb{R}^m$ est linéaire si et seulement si $f(0) = 0$.*

1.1.2 Notions de convexité

Définition 1.1.5. *1. Un ensemble D de $\mathbb{R}^n$ est dit convexe si*

$$\forall x, \; y \in D : \; tx + (1-t)y \in D, \quad \forall t \in [0,1]$$

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

$$D = \{x \in \mathbb{R}^n : \; A_i^t x \leq b_i, \; i = \overline{1,m}\}$$

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

3. *S_n est un (n-simplexe) s'il est de la forme*

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

4. *Une fonction f est convexe sur un ensemble convexe D si*

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y),\ \forall t \in [0,1]$$

5. *Une fonction f est strictement convexe sur un ensemble convexe D si*

$$f(tx + (1-t)y) < tf(x) + (1-t)f(y),\ \forall t \in]0,1[\text{ et } x \neq y$$

Définition 1.1.6. ***(Combinaison linéaire convexe)***
Un vecteur $y \in D \subset \mathbb{R}^n$ est une combinaison linéaire convexe des points $\{x_1, \ldots, x_p\}$ s'il existe des coefficients réels $\lambda_i \geq 0$, $i \in \{1, \ldots, p\}$, tels que :

$$y = \sum_{i=1}^{p} \lambda_i x_i \quad avec \sum_{i=1}^{p} \lambda_i = 1.$$

Définition 1.1.7. ***(Enveloppe convexe)***
L'enveloppe convexe d'un ensemble $S \subset \mathbb{R}^n$, est l'ensemble des points de $\mathbb{R}^n$ qui s'écrivent comme combinaison convexe des points de S.
Elle est notée par :

$$conv(S) = \{x \in \mathbb{R}^n : \ x = \sum_{i=1}^{p} \lambda_i x_i,\ x_i \in S,\ \lambda_i \geq 0,\ \forall i = \overline{1,p} \text{ et } \sum_{i=1}^{p} \lambda_i = 1\}.$$

Définition 1.1.8. ***(Point extrême)***
Soit D un convexe non vide de $\mathbb{R}^n$, x est dit point extrême ou sommet de D si

$$x = \lambda x_1 + (1-\lambda)x_2 \quad \forall x_1,\ x_2 \in D \text{ et } \lambda \in]0,1[\quad alors \quad x = x_1 = x_2$$

Tout point extrémal d'un convexe D est un point de la frontière de D.

Proposition 1.1.5. *Soit D un convexe non vide de $\mathbb{R}^n$, alors*
x est un sommet de D si et seulement si $D \backslash \{x\}$ est un convexe.

Définition 1.1.9. ***(Intérieur relatif)*** *L'intérieur relatif d'un sous ensemble non vide D de $\mathbb{R}^n$, noté $ri(D)$ est défini comme l'intérieur de D vu comme sous-ensemble de $aff(D)$.*

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

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

Théorème 1.1.6. *(**Caractérisation de** $ri(D)$)*
Soit D un convexe non vide de $\mathbb{R}^n$. Alors

$$ri(D) = \{z \,/\, \forall x \in D, \exists \mu > 1 : (1-\mu)x + \mu z \in D\}.$$

Autrement dit, tout segment de droite dans D ayant z comme extrémité, peut être prolongé au-delà de z sans sortir de D.

1.1.3 Convexité et dérivée

Définition 1.1.10. *Le gradient d'une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ continument différentiable évalué au point $x \in \mathbb{R}^n$ s'écrit*

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

et les coefficients de la ième ligne et la jème colonne de la matrice Hessienne s'écrivent

$$\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \ i, j = \overline{1, n}.$$

Proposition 1.1.7. *Si $f \in C^1(C)$, où C est un ensemble convexe, alors on a les équivalences suivantes :*

- *f est convexe si et seulement si*

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

- *f est convexe si et seulement si*

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

De plus, f est dite strictement convexe si l'une ou l'autre des inégalités précédentes sont strictes pour $x \neq y$.

Proposition 1.1.8. *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction continument différentiable sur un domaine convexe D.*

1. *f est une fonction convexe sur D si et seulement si la matrice Hessienne $\nabla^2 f(x)$ est semi-définie positive, c'est-à-dire*

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

ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont positives.

2. *f est une fonction strictement convexe sur D si et seulement si la matrice Hessienne $\nabla^2 f(x)$ est définie positive, c'est-à-dire*

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

ou encore toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives.

Lemme 1.1.9. *Une fonction f est dite concave si $-f$ est convexe.*

Définition 1.1.11. *Une fonction f est dite cœrcive sur un ensemble convexe S si*

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

Définition 1.1.12. *On appelle sous-gradient d'une fonction f convexe sur S au point $x_0 \in S$ tout vecteur $r = (r_1, r_2, ..., r_n)^T \in \mathbb{R}^n$ vérifiant :*

$$f(x) \geq f(x_0) + r^T (x - x_0), \forall x \in S.$$

Définition 1.1.13. *On appelle sous-différentiel de f au point x_0 l'ensemble de tous les sous-gradients de f en x_0. On le note $\partial f(x_0)$.*

Définition 1.1.14. *(**Direction admissible**)*
Soit C une partie non vide de $\mathbb{R}^n$. Un vecteur $d \neq 0$ de $\mathbb{R}^n$ est dit direction de C en $a \in C$ si

$$\exists \overline{\alpha} > 0 \,/\, a + \alpha d \in C, \forall \alpha \in [0, \overline{\alpha}].$$

Si la propriété est vraie $\forall a \in C$, on dit que d est une direction admissible pour C.

Définition 1.1.15. *(**Dérivée directionnelle**)*
*Soit f une fonction quelconque de $\mathbb{R}^n$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$ et x^0 un point dans $\mathbb{R}^n$ tel que $f(x^0)$ est finie. Alors la dérivée directionnelle (**unilatérale**) de f en x^0 dans la direction $d \in \mathbb{R}^n$ est définie par*

$$f'(x^0, d) = \lim_{\lambda \to 0^+} \frac{f(x^0 + \lambda d) - f(x^0)}{\lambda} \text{ si elle existe.}$$

($\pm\infty$ sont des valeurs admises pour la limite).

Notons que

$$-f'(x^0, -d) = \lim_{\lambda \to 0^-} \frac{f(x^0 + \lambda d) - f(x^0)}{\lambda}.$$

- La dérivée directionnelle est dite **bilatérale** si $f'(x^0, d) = -f'(x^0, -d)$.
- Si $f'(x^0, d)$ existe $\forall d \in \mathbb{R}^n$, on dira que f est directionnellement différentiable en x^0.
- S'il existe un vecteur constant $g_0 \in \mathbb{R}^n$ tel que $f'(x^0, d) = \langle g_0, d \rangle,\ \forall d \in \mathbb{R}^n$, on dit que f est **G-différentiable** (ou différentiable au sens de Gateaux) en x^0 et on écrit : $f'(x^0) = g_0$.

Théorème 1.1.10. *(**Séparation large**)*
Soit $H = \{x \in \mathbb{R}^n :\ a^t x = \alpha\}$. $C_1,\ C_2$ deux sous ensembles non vides de $\mathbb{R}^n$ sont séparés au sens large par H si

$$a^t x \geq \alpha,\ \forall x \in C_1 \text{ et } a^t x \leq \alpha,\ \forall x \in C_2$$

Théorème 1.1.11. *(**Séparation stricte**)*
Soit $H = \{x \in \mathbb{R}^n :\ a^t x = \alpha\}$. C_1, C_2 deux sous ensembles non vides de $\mathbb{R}^n$ sont strictement séparés par H si

$$a^t x > \alpha,\ \forall x \in C_1 \text{ et } a^t x < \alpha,\ \forall x \in C_2$$

Remarque 1.1.1. *On peut toujours séparer strictement un point y d'un ensemble convexe fermé non vide C ne contenant pas y.*

Lemme 1.1.12. *Soit C un convexe fermé non vide de $\mathbb{R}^n$ et $y \in (\mathbb{R}^n / C)$. Alors il existe $a \in \mathbb{R}^n,\ a \neq 0$ et un scalaire $\alpha \in \mathbb{R}$ tels que*

$$a^t y > \alpha \ \text{ et } \ a^t x \leq \alpha \ \ \forall x \in C$$

Définition 1.1.16. *Un sous ensemble D est un cône si et seulement si*

$$\forall x \in D,\ \forall \lambda > 0 :\ \lambda x \in D$$

Un cône est donc une union de demi droites fermées positives issues de l'origine, ce dernier peut (ou ne peut pas) appartenir à D. Si D est convexe, on dit qu'il est un cône convexe.

Définition 1.1.17. *Un cône D est dit pointé ou saillant ssi $D \cap (-D) = \{0\}$.*

Définition 1.1.18. *Soit f une fonction définie par $f : \mathbb{R}^m \longrightarrow \mathbb{R} \cup \{\infty\}$, on dit que f est inf-compact si pour tout $r > 0$, l'ensemble $S_r(f) = \{x \in \mathbb{R}^m :\ f(x) \leq r\}$ est compact dans $\mathbb{R}^m$.*

Proposition 1.1.13. *Soit C un convexe non vide de $\mathbb{R}^n$, et $a \in C$, on pose*
$C_\infty(a) = \{d \in \mathbb{R}^n / \ a + \lambda d \in C, \ \forall \lambda > 0\}$,
alors $C_\infty(a)$ est un cône convexe non vide.

Définition 1.1.19. *On appelle cône de récession (ou asymptote) de C l'ensemble*

$$C_\infty = \bigcap_{a \in C} C_\infty(a).$$

Un élément $d \in C_\infty$, est appelé direction de récession.

Proposition 1.1.14. *Soit C un convexe, fermé et non vide de $\mathbb{R}^n$, alors*

- *$C_\infty(a) = C_\infty(b), \ \forall a, b \in C$.*
- *L'ensemble C_∞, est un cône convexe fermé*
- *C est borné si et seulement si $C_\infty = \{0\}$.*

1.1.4 Fonction barrière

Définition 1.1.20. *Soit $C \subset \mathbb{R}^n$, on appelle fonction barrière associée à C toute fonction B définie sur $int(C)$ telle que*

$$\begin{cases} B \text{ est continue} \\ B(x) \geq 0, \forall x \in int(C) \\ \lim B(x) = +\infty \ \text{ quand } \ x \to x^* \in Fr(C) \end{cases}$$

La fonction barrière classique utilisée en programmation mathématique est la fonction barrière logarithmique.

1.1.5 Programmation mathématique

La programmation mathématique constitue un domaine vaste et riche dans le domaine d'optimisation et dans l'analyse numérique, et elle traduit plusieurs problèmes pratiques importants.

Définition 1.1.21. *Un programme mathématique (PM) est un problème d'optimisation écrit sous la forme*

$$(PM) \begin{cases} \min f(x) \\ \text{sous les contraintes} \\ D = \{x \in \Omega \subset \mathbb{R}^n : g_i(x) \leq 0, \ i = \overline{1,n} \ \text{ et } \ h_j(x) = 0, \ j = \overline{1,m}\}. \end{cases}$$

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

- *Un point x de D est appelé solution réalisable de (PM).*
- *D est l'ensemble des solutions réalisables de (PM).*
- *On appelle solution optimale de (PM), toute solution réalisable x^* réalisant le minimum de f sur D, et $f(x^*)$ sera appelée valeur optimale de (PM).*

La classification de (PM) et son traitement numérique sont établis à partir des propriétés fondamentales des fonctions f, g_i et h_j à savoir la convexité, la différentiabilité et la linéarité. Parmi les cas particuliers les plus étudiés on note

- La programmation linéaire (f linéaire, g_i, h_j affine, D ortant positif).
- La programmation convexe (f, g_i, convexe, h_j affine, D convexe).
- La programmation en nombres entiers (D discret).

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

Définition 1.1.22. *Une solution réalisable x^* est un minimum local de (PM) s'il existe $\varepsilon > 0$ tel que $f(x^*) \leq f(x), \forall x \in \mathbf{B}(x^*, \varepsilon)$ où $\mathbf{B}(x^*, \varepsilon) = \{x \in \mathbb{R}^n : \| x - x^* \| \leq \varepsilon\}$.*

a) Qualification des contraintes :

Une contrainte g_i est dite active (ou saturée) en $x^* \in D$ si $g_i(x^*) = 0, \forall i = (1, \ldots, n)$.

Un point $x^* \in D$ est dit régulier (on dit également que les contraintes sont qualifiées en x^*) si les gradients des contraintes saturées en x^* sont linéairement indépendants.

Il existe aussi deux critères usuels de qualification en tout point de D, à savoir :

▪ Si toutes les contraintes sont affines.

▪ Si D est défini uniquement par des inégalités, on a le critère de Slater suivant : si $g_i(\overline{x})$ est convexe pour tout $i = \overline{1, n}$ et qu'il existe un point $x^0 \in D$ tel que $g_i(x^0) < 0$, $(int(D) \neq \emptyset)$.

b) Conditions nécessaires d'optimalité

Théorème 1.1.15. ***(Karush-Kuhn-Tucker)***
Soit $x^* \in D$ satisfaisant l'une des conditions de qualification et supposons que f, g_i, h_j sont $C^1(\mathbb{R}^n)$, on a :
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

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{m} y_j \nabla h_j(x^*) = 0 \\ \lambda_i g_i(x^*) = 0, \ i = \overline{1,n} \\ h_j(x^*) = 0, \ j = \overline{1,m}. \end{cases}$$

Si de plus f, g_i, h_j sont convexes, les conditions d'optimalité précédentes sont à la fois nécessaires et suffisantes pour que x^* soit un optimum global pour (PM). Les problèmes avec contraintes de type (PM) peuvent souvent êtres transformés en problèmes sans contraintes à l'aide des multiplicateurs de Karush Kuhn Tucker (KKT). Cette méthode qui nous permet de trouver les point stationnaires x^* de la fonction f, revient en effet à associer à chaque contrainte un scalaire inconnu λ_i, y_j. On forme par la suite une combinaison linéaire des gradients de la fonction objectif et des contraintes comme le montre le système ci-dessus.

Dans le cas particulier où $n = 0$, les conditions précédentes sont appelées les conditions de Lagrange et s'écrivent comme suit

Si x^* est un optimum local, alors il existe $\lambda_j \in \mathbb{R}$, $j = \overline{1,m}$ tel que

$$\begin{cases} \nabla f(x^*) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x^*) = 0 \\ h_j(x^*) = 0, \ j = \overline{1,m}. \end{cases}$$

c) Fonction de Lagrange et dualité lagrangienne classique

On considère le programme de type suivant

$$(P_g)\begin{cases} \min f(x) \\ g_i(x) \leq 0,\ i = \overline{1,m} \\ x \in S \subset \mathbb{R}^n : \end{cases}$$

Associons à chaque contrainte un nombre réel $\lambda_i > 0$ appelé multiplicateur de Lagrange.

La fonction de Lagrange (où lagrangien) associée à ce problème est définie par

$$L(x,y) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x),\ \ x \in S,\ \lambda_i > 0.$$

On appelle (x^*, λ^*) un point selle (ou point col) pour la fonction de Lagrange sur $S \times \mathbb{R}^m_+$, si tout point $(x, \lambda) \in S \times \mathbb{R}^m_+$ on a

$$\min_{x\in S} L(x,\lambda) = L(x^*,\lambda) \leq \min_{x\in S} \max_{\lambda\in\mathbb{R}^m_+} L(x,\lambda) = L(x^*,\lambda^*) \leq \max_{\lambda\in\mathbb{R}^m_+} L(x,\lambda) = L(x,\lambda^*).$$

Rappelons une propriété caractéristique des points selles

(x^*, λ^*) est un point selle si et seulement si :

- $L(x^*,\lambda^*) = \min_{x\in S} L(x,\lambda^*)$
- $g_i(x^*) \leq 0,\ i = \overline{1,m}.$
- $\lambda_i^* g_i(x^*) = 0,\ i = \overline{1,m}.$

De plus on a le résultat suivant, si (x^*, λ^*) est un point selle pour la fonction de Lagrange, alors x^* est un minimum global pour (P_g).

Définissons maintenant pour $\lambda > 0$ la fonction q donnée par

$$q(\lambda) = \min_{x\in S} L(x,\lambda)$$

Et le programme

$$(D_g)\begin{cases} \max q(\lambda) = \max\limits_{\lambda\in\mathbb{R}^m_+} \min L(x,\lambda) \end{cases}$$

La fonction q est appelée la fonction duale et (D_g) le programme dual associé au programme primal (P_g).

Revenons à la programmation mathématique linéaire qui sera un outil important pour démontrer ultérieurement un résultat fondamental.

Dualité lagrangienne

Soit

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

et on considère le problème d'optimisation

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

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

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

On pose

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

donc

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

Le problème dual est

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

On a l'inégalité de dualité

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

1.2 Programmation linéaire

La programmation linéaire dans $\mathbb{R}^n$ traite la résolution des problèmes d'optimisation pour lequels la fonction objectif, et les contraintes sont linéaires. Les spécialistes du domaine, la considèrent comme étant la technique la plus utilisée dans la recherche opérationnelle.

Définition 1.2.1. *Un programme linéaire (PL) est un système d'équations ou d'inéquations appelées "contraintes" qui sont linéaires et à partir de ses contraintes on doit optimiser une fonction également linéaire appelée "objectif" (économique) sous la forme*

$$(PL)\begin{cases} Opt(\sum_{j=1}^{n} c_j x_j) \\ sous\ les\ contraintes \\ \sum_{j=1}^{n} a_{ij}x_j(\leq,=,\geq)b_i,\ \ i=\overline{1,m} \\ x_j \geq 0,\ \ j=\overline{1,n}. \end{cases}$$

(PL) *s'écrit aussi sous la forme matricielle*

$$(PL)\begin{cases} Opt\ (c^t x) \\ sous\ les\ contraintes \\ Ax(\leq,=,\geq)b \\ x \geq 0. \end{cases}$$

Tel que

- $Opt = (\max\ ou\ \min)$.
- $c^t x = \sum_{j=1}^{n} c_j x_j$ *est la fonction à optimiser.*
- $c \in \mathbb{R}^n$ *est le vecteur coût.*
- $b \in \mathbb{R}^m$ *est le second membre.*
- $A = (a_{ij})_{\substack{1\leq i\leq m \\ 1\leq j\leq n}} \in \mathbb{R}^{m\times n}$ *est la matrice des contraintes.*

Exemple 1.2.1.

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

1.3 Formes usuelles d'un programme linéaire

Les programmes linéaires (PL) se présentent sous deux formes différentes :

1.3.1 Forme standard

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

$$(PL)\begin{cases} Opt\ (c^t x) \\ sous\ les\ contraintes \\ Ax = b \\ x \geq 0. \end{cases}$$

1.3.2 Forme canonique

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

$$(PL)\begin{cases} Opt\ (c^t x) \\ sous\ les\ contraintes \\ Ax(\geq ou \leq)b \\ x \geq 0. \end{cases}$$

Remarque 1.3.1. *On peut toujours mettre un programme linéaire quelconque sous forme standard en introduisant des variables supplémentaires positives appelées "variables d'écart".*

Exemple 1.3.1.

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

En introduisant les variables d'écart $x_3 \geq 0,\ x_4 \geq 0$ *dans la première et la deuxième contrainte, on met le problème précédent sous la forme standard suivante*

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

Théorème 1.3.1. *L'ensemble réalisable du problème de programmation linéaire (PL) est un polyèdre convexe.*

1.4 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, on peut toujours lui associer un autre problème appelé dual (noté par (DL)).

Soit le problème primal

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

Son dual s'écrit sous la forme

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

En effet, la fonction de Lagrange de (PL) est

$$L(x,y) = c^t x + y^t(b - Ax), y \in \mathbb{R}^m$$

qui peut s'écrire aussi sous la forme

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

Donc

$$\min_{x\in\mathbb{R}^m_+} L(x,y) = \begin{cases} b^t y\ si\ (c^t - y^t A) \geq 0 \\ -\infty\ \ sinon \end{cases}$$

D'où le dual de (PL) est $\max\limits_{y\in\mathbb{R}^m} \min\limits_{x\in\mathbb{R}^m_+} L(x,y)$ défini par

$$(DL)\begin{cases} \max\ \ b^t y \\ c - A^t y \geq 0. \end{cases}$$

Le problème linéaire (DL) est très lié au problème linéaire (PL), on remarque en effet,

• Que la matrice des contraintes de (DL) est la transposée de la matrice des contraintes de (PL).

• Que le vecteur coût de (PL) est le vecteur de second membre des contraintes de (DL).

• Que le vecteur coût de (DL) est le vecteur de second membre des contraintes de (PL).

Définition 1.4.1. ***(Définition du dual dans le cas général)***
Evidemment, on peut définir le dual d'un problème linéaire quelconque (pas nécessairement sous forme standard), le tableau suivant résume les correspondances entre le problème primal et son dual et permet d'écrire directement le dual d'un problème linéaire quelconque.

Primal	Dual
Fonction objectif (min)	Second membre
Second membre	Fonction objectif (max)
A matrice des contraintes	A^t matrice des contraintes
Contrainte i $\geq$	Variable $y_i \geq 0$
Contrainte $i =$	Variable $y_i \lesseqgtr 0$
Variable $x_j \geq 0$	Contrainte j $\leq$
Variable $x_j \lesseqgtr 0$	Contrainte $j =$

Exemple 1.4.1. *Soit le problème primal suivant*

$$(PL)\begin{cases} \min(100x_1 + 600x_2) \\ 2x_1 + 5x_2 = 3 \\ 2x_1 + 2x_2 = 2 \\ 4x_1 + 6x_2 = 1 \\ x_1,\ x_2 \geq 0. \end{cases}$$

Alors son dual est

$$(DL)\begin{cases} \max(3y_1 + 2y_2 + y_3) \\ 2y_1 + 2y_2 + 4y_3 \leq 100 \\ 5y_1 + 2y_2 + 6y_3 \leq 600 \\ (y_1,\ y_2,\ y_3) \in \mathbb{R}^3. \end{cases}$$

Remarque 1.4.1.

1. *Le dual du problème dual (DL) est le problème primal (PL).*
2. *On peut transformer un problème de maximisation à un problème de minimisation. Il suffit d'écrire* $\max f = -\min(-f)$.

Remarque 1.4.2. *Dans notre travail, on s'intéresse à la résolution des problèmes linéaires sous forme standard de type*

$$(DL)\begin{cases} \min b^t y \\ A^t y \geq c \\ y \in \mathbb{R}^m. \end{cases}$$

Théorème 1.4.1. ***(Dualité faible)***
Si x et y sont des solutions réalisables de problèmes (PL) et (DL) respectivement, alors $c^t x > b^t y$.

Preuve.

On a x et y sont réalisables, alors du problème (DL) on a :

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

■

Théorème 1.4.2. ***(Dualité forte)***
Si x^* et y^* sont des solutions réalisables de (PL) et (DL) respectivement tels que $b^t y^* = c^t x^*$, alors x^* et y^* sont des solutions optimales des problèmes (PL) et (DL) respectivement.

Preuve.

On veut démontrer que x^* et y^* sont des solutions optimales des problèmes (PL) et (DL) respectivement c'est-à-dire, on démontre que $c^t x^* = \min c^t x$ et $b^t y^* = \max b^t y$.

D'après le théorème précédent on a $c^t x \geq b^t y$ donc $\min c^t x \geq \max b^t y$

en particulier, $c^t x^* \geq \min c^t x \geq \max b^t y \geq b^t y^*$.

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

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

D'où le résultat. ■

Chapitre 2

Méthodes de résolution d'un programme linéaire

Dans ce chapitre, on présente quelques domaines d'applications de la programmation linéaire ainsi que sa résolution par des méthodes numériques. Nous mettons l'accent sur deux grands types de méthodes, à savoir : les méthodes simpliciales et les méthodes de points intérieurs.

2.1 Domaines d'applications de la programmation linéaire

La programmation linéaire (PL) est une technique mathématique permettant de déterminer la meilleure solution d'un problème dont les données et les inconnus satisfont à une série d'équations et d'inéquations linéaires. La programmation linéaire à été formulée par Dantzig en 1947 et connaît un développement rapide par suite de son application directe à la gestion scientifique des entreprises. Le facteur expliquant l'essor de la programmation linéaire est la construction d'ordinateurs puissants qui ont permis de traiter les problèmes concrets de taille très grande. On l'applique surtout en gestion et on 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.

La gestion de production est le domaine où ces applications sont les plus nombreuses. On citera entre-autres :

• l'élaboration de plans de production et de stockage,

• le choix de techniques de production,

• l'affectation de moyens de production,

• la détermination de la composition de produits.

Les applications sont également nombreuses dans le domaine du marketing avec, en particulier :

• le choix de plan-média,

• la détermination des politiques de prix,

• la répartition des efforts de la force de vente,

• la sélection des caractéristiques du produit.

La technique de la programmation linéaire est couramment appliquée dans l'industrie pétrolière. C'est l'une des industries, si ce n'est la principale qui utilise quotidiennement la programmation linéaire. Elle est l'outil qui permet aux raffineurs de faire la détermination optimale de production d'une raffinerie. Pour ce faire, le programme doit tenir compte d'un certain nombre de contraintes telles que :

• bruts disponibles, leurs rendements et les qualités des coupes,

• spécifications des produits à fabriquer,

• limitation de débouchés pour certains produits,

• capacités des unités,

• modes de réglages des installations,

• capacités de stockages disponibles.

La programmation linéaire peut également être utilisée dans d'autres domaines de raffinage, par exemple :

• calcul de la composition optimale des mélanges de produits (carburants, gasoils, fuels) en tenant compte des spécifications.

• l'optimisation dans l'utilisation des installations,

• calcul de l'obtention du meilleur préchauffage des bruts et des charges,

• détermination du meilleur équilibre (vapeur-éléctricité) d'une raffinerie.

On peut utiliser aussi la programmation linéaire dans la recherche opérationnelle pour par exemple optimiser le fonctionnement d'une flotte de tankers et la mise en place des produits.

Les méthodes de résolution sont nombreuses parmi lesquelles la méthode du simplexe et des méthodes récentes appelées méthodes de points intérieurs. Avant de présenter ces méthodes d'une manière détaillée, on donne quelques exemples concrets et faciles qui peuvent se modéliser sous forme d'un programme linéaire.

2.1.1 Exemples concrets de problèmes de programmation linéaire

Problème de transport

Une entreprise stocke un produit dans trois dépôts A_1, A_2, A_3. Les quantités stockées sont respectivement a_1, a_2, a_3. Les dépôts doivent alimenter quatre magasins de vente B_1, B_2, B_3 et B_4. La quantité du produit nécessaire au point de vente B_i est b_i, $i = \overline{1,4}$. Comment l'entreprise doit-elle répartir les stocks entre les points de vente ou quelle quantité le dépôt A_i doit-il livrer au point de vente B_j ?.

Le problème de transport est un problème particulier de la programmation linéaire. Sa formulation mathématique est :

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$\sum_{i=1}^{m} a_i \geq \sum_{j=1}^{n} b_j$. Cette équation traduit que la demande doit être satisfaite.

$$\sum_{j=1}^{n} x_{ij} \leq a_i, \quad i = \overline{1,m}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}$$

$$x_{ij} \geq 0,\ i = \overline{1,m}\ et\ j = \overline{1,n}.$$

$a_i \geq 0,\ i = \overline{1,m},\ b_j \geq 0,\ j = \overline{1,n},\ c_{ij} \geq 0,\ i = \overline{1,m},\ j = \overline{1,n}.$

Problème de répartition de ressources

Une entreprise peut produire différents biens. Chaque bien repéré par un indice "j". Pour réaliser sa production, elle utilise des matières premières, des machines, de la main d'oeuvre, c'est à dire des ressources mesurées en unités appropriées. Ces ressources, ou facteurs de production, sont disponibles en quantité limitée ; soit b_i la quantité disponible de la ressource i. On sait par ailleurs, quelle quantité a_{ij} la production d'une unité du bien j nécessite une ressource i.

Supposons que la production soit à rendement constant, c'est à dire que la production de x_j unités du bien j exige $a_{ij}x_j$ unités de la ressource i.

Ce programme s'écrit sous la forme

$$\max z = \sum_{j=1}^{n} c_j x_j \ /\ \sum_{j=1}^{n} a_{ij}x_j \leq b_i,\ \ i = \overline{1,m}.\ x_j \geq 0,\ j = \overline{1,n}.$$

Problème de mélange

Il faut mélanger trois gaz de telle manière que le gaz mixte soit bon marché et qui possède un pouvoir calorifique entre plus de 1700 et 2000 $K.cal/m^3$ et un taux de sulfure au plus de 2,8 g/m^3.

Indication sur les trois gaz :

gaz	*pouvoir calorifique en* $K.cal/m^3$	*taux de sulfure en* g/m^3	*prix en DA*
1	1000	6	100
2	2000	2	250
3	1500	3	200

Le modèle mathématique de ce problème s'écrit comme suit :

Soient x_1, x_2 et x_3 le nombre de m^3 à fabriquer respectivement du premier, second et troixième gaz.

Ce programme s'écrit sous la forme suivante

$$\begin{cases} \min z = 100x_1 + 250x_2 + 200x_3 \\ 1000x_1 + 2000x_2 + 1500x_3 \leq 2000 \\ 1000x_1 + 2000x_2 + 1500x_3 \geq 1700 \\ 6x_1 + 2x_2 + 3x_3 \leq 2,8 \\ x_1,\ x_2,\ x_3 \geq 0. \end{cases}$$

Problème de nutrition

On se propose de fournir quotidiennement et à chaque individu d'une population un minimum de 70 g de protéines, 300 unités de calories, 800 mg de calcium et 12 mg de fer. Les produits disponibles sont le pain, le beurre, le fromage, les pois et les épinards. Les prix par 100 g de ces produits sont respectivement 5, 34, 40, 10 et 5 DA. Le problème est de constituer, aux moindres frais, les raisons quotidiennes respectant les exigences du régime imposé.

Les quantités de protéines (en g), de calories (en unités), de calcium (en mg) et de fer (en mg) par 100 g de ces aliments sont données dans le tableau suivant :

	Proteines	*Calories*	*Calcium*	*Fer*
Pain	10	300	50	4
Beurre	30	1800	400	–
Fromage	35	800	450	–
Pois	20	1500	750	4
Epinards	25	300	120	15

Soit x_1 le nombre de pain introduit dans la ration de 100 g

x_2 le nombre de beurre introduit dans la ration de 100 g

x_3 le nombre de fromage introduit dans la ration de 100 g

x_4 le nombre de pois introduit dans la ration de 100 g

x_5 le nombre d'épinards introduit dans la ration de 100 g.

Donc on obtient le problème de programmation linéaire suivant

$$\begin{cases} \max 3x_1 + 7x_2 + 7x_3 + 5x_4 + 5x_5 \\ 10x_1 + 30x_2 + 35x_3 + 20x_4 + 25x_5 \geq 70 \\ 300x_1 + 1800x_2 + 800x_3 + 1500x_4 + 300x_5 \geq 300 \\ 50x_1 + 400x_2 + 450x_3 + 750x_4 + 120x_5 \geq 800 \\ 4x_1 + 4x_4 + 15x_5 \geq 12 \\ x_i \geq 0, \ i = \overline{1,5}. \end{cases}$$

Programme de raffinerie

Une raffinerie peut traiter trois types de pétroles bruts appelés brut1, brut2, brut3 originaires de trois pays différents. Par distillation fractionnée dans des "toppings". Ces pétroles bruts donnent des coupes qui sont des ensembles d'hydrocarbures ayant des températures d'ébullition comprises entre des limites fixées. Un "topping" permet d'obtenir : des gaz et des gaz liquéfiés.

Une gazoline ou coupe $0-80$°c (hydrocarbures donc le point d'ébullition est compris entre 0 et 80°c)

Une benzine ou coupe $80 - 130$°c

Une naphta léger ou coupe $130 - 160$°c

Une naphta lourd ou coupe $160 - 190$°c

Une kérosène ou coupe $190 - 230$°c

Une gazoil léger ou coupe $230 - 310$°c

Une gazoil lourd ou coupe $310 - 400°c$

Un fuel-oil ou coupe $> 400°c$

Ces coupes subissent ensuite des traitements complémentaires (épuration, désulfuration, cracking, reforming catalytique) pour devenir des bases qui, convenablement mélangées, permettront d'obtenir les produits finis : des gaz et gaz liquéfiés, des essences, du pétrole, du gasoil et du fuel-oil. Les rendements des pétroles bruts traités sont précisés par le tableau ci-après (qui explicite les quantités produites à partir de 100 tonnes de brut) :

Matière première production	$brut1$ $(tonnes)$	$brut2$ $(tonnes)$	$brut3$ $(tonnes)$
gaz et gaz liquéfiés	2	6	6
Essences	20	25	30
Pétroles	8	25	30
Fuel-oil	30	50	30
Total	100	100	100

La raffinerie peut produire au maximum, au cours d'une année : 300000 tonnes de gaz et gaz liquéfiés, 1050000 tonnes d'essences, 180000 tonnes de pétroles, 1350000 tonnes de gasoil et 1800000 tonnes de fuel-oil.

Elle réalise un bénéfice de 4000 DA par tonne de brut1 mise en oeuvre, 5000 DA par tonne de brut2 et 6000 DA par tonne de brut3 mise en oeuvre.

Quelle quantité de chacun des bruts devera-t-elle traiter pour réaliser le bénéfice total maximal ?

Le problème répondant à cette question se modélise sous forme d'un programme linéaire comme suit :

Appelons respectivement x_1, x_2 et x_3 les quantités de brut, en millions de tonnes, traitées annullement par la raffinerie. Le tableau de rendements ci-dessus montre que la produc-

tion de gaz et gaz liquéfiés correspondant à 1 million de tonnes de pétroles brut atteint :

0,02 milion de tonnes quand on traite du brut $n°1$

0,06 milion de tonnes quand on traite du brut $n°3$

Comme la fabrication de cette catégorie de produit est limitée à 300000 tonnes, soit 0,3 million de tonnes, les contraintes correspondantes s'écrivent

$$0,2x_1 + 0,6x_3 \leq 0,3$$

Soit encore

$$x_1 + 3x_3 \leq 15$$

On obtient de même, pour limitation de production d'essences :

$$0,02x_1 + 0,25x_2 + 0,30x_3 \leq 1,05$$

soit encore

$$4x_1 + 2x_3 \leq 9$$

Pour la limitation de production pétrole :

$$0,08x_1 + 0,04x_3 \leq 0,18$$

soit encore

$$4x_1 + 2x_3 \leq 9$$

Pour la limitation de production de gasoil :

$$0,40x_1 + 0,25x_2 + 0,30x_3 \leq 1,35$$

qui est équivalent à l'équation :

$$8x_1 + 5x_2 + 6x_3 \leq 27$$

Pour la limitation de production de fuel-oil :

$$0,30x_1 + 0,50x_2 + 0,30x_3 \leq 1,80$$

soit encore

$$3x_1 + 5x_2 + 3x_3 \leq 18$$

Le problème est de maximiser le bénéfice en millions de DA, qui s'écrit sous la forme suivante

$$\begin{cases} \max 40x_1 + 50_2 + 60x_3 \\ x_1 + 3x_3 \leq 15 \\ 4x_1 + 5x_2 + 6x_3 \leq 21 \\ 4x_1 + 2x_3 \leq 9 \\ 8x_1 + 5x_4 + 6x_3 \leq 27 \\ 3x_1 + 5x_2 + 3x_3 \leq 18 \\ x_1,\ x_2,\ x_3 \geq 0. \end{cases}$$

Problème des ordures ménagères

Une ville A produit quotidiennement 500 tonnes d'ordures ménagères, une ville B produit quotidiennement 400 tonnes. Les ordures doivent être incinérées à l'incinérateur 1 ou à l'incinérateur 2 qui peuvent traités respectivements jusqu'à 500 tonnes par jour. Le coût de l'incinérateur des ordures est de 40 DA par tonne à l'incinérateur 1, et 30 DA la tonne à l'incinérateur 2.

L'incinération des ordures réduit chaque tonne de déchets à 0,2 tonnes de débris, qui seront enfouis dans deux terrains vagues.

Chaque terrain vague peut recevoir au plus 200 tonnes de débris par jour. Il coûtera 3 DA le kilomètre pour transporter une tonne de déchets (ordures ou débris). Les distances entre les différents lieux sont données par le tableau ci dessous. Ce problème peut se formuler comme un problème de programmation linéaire pour minimiser le coût total de prélèvement des ordures ménagères des deux villes A et B.

En effet :

	$Incinérateur1$	$Incinérateur2$
Ville A	30	5
Ville B	36	42
terrain vague 1	5	9
Terrain vague 2	8	6

Soient x_{ij} le nombre de tonnes de déchets à transporter de la ville i, $i = \overline{1,2}$ à l'incinérateur j et y_{jk} le nombre de tonnes de débris à transporter de l'incinérateur j au terrain-vague k ($k = \overline{1,2}$).

Donc le programme correspondant est

$$\begin{cases} \min(40(x_{11}+x_{21})+30(x_{12}+x_{22})+3(30x_{11}+5x_{12}+36x_{21}+42x_{22} \\ \quad +5y_{11}+9y_{12}+8y_{21}+6y_{22})) \\ x_{11}+x_{12}=500 \\ x_{21}+x_{22}=400 \\ y_{11}+y_{21}=0,2(x_{11}+x_{21}) \\ y_{21}+y_{22}=0,2(x_{12}+x_{22}) \\ x_{11}+x_{21}\leq 500 \\ x_{12}+x_{22}\leq 500 \\ y_{11}+y_{21}\leq 200 \\ y_{12}+y_{22}\leq 200 \\ x_{ij}\geq 0,\ \ y_{jk}\geq 0,\ (i,\ j,\ k=\overline{1,2}). \end{cases}$$

Dans la partie suivante, on présente des méthodes numériques pour résoudre un problème de la programmation linéaire, en proposant des méthodes typiques, telle que la méthode de simplexe, et celle de points intérieurs.

2.2 Méthode de simplexe

Elle a été développé à la fin des années 40 par G. Dantzig. 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'excédant pas le nombre de sommets $c_n^m = \frac{n!}{m!(n-m)!}$ 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 de part sa complexité arithmétique exponentielle qui est de l'ordre de $O(2^n)$ opérations.

2.2.1 Caractéristiques de l'algorithme du simplexe

On peut dégager les remarques et les propriétés suivantes

1. Cet algorithme nécessite un point de départ admissible (réalisable). Cependant, il existe une méthode appelée simplexe à deux phases permettant de résoudre le problème où l'on ne dispose pas d'un tel point de départ.
2. Cet algorithme ne permet pas de traiter naturellement les variables libres, c'est-à-dire des variables sans contraintes de positivité. Il faut alors décomposer ces variables ce qui augmente la taille du vecteur x et de la matrice A.
3. Cet algorithme est sensible à la dégénérescence qui diminue ses performances et peut entraîner des problèmes de cyclage. Dans les applications réelles, presque tous les problèmes sont dégénérés.
4. Bien que très efficace en pratique, cette méthode ne satisfait pas les théoriciens à

cause de sa complexité exponentielle.

Cela signifie qu'il existe des cas où le nombre d'itérations nécessaire pour atteindre la solution est exponentiel en fonction de la taille du problème (n). On cite l'exemple de Klee et Minty (1972) qui montrait pour la première fois que la méthode du simplexe pouvait prendre un nombre exponentiel d'itérations et qu'elle visite tous les sommets du polyèdre (ensemble des contraintes de l'exemple considéré).

2.3 Méthodes de points intérieurs

Les méthodes de points intérieurs comme leur non l'indique éviteront soigneusement la frontière de l'ensemble admissible et ne souffriront donc pas de l'aspect combinatoire inhérent à la méthode du simplexe. Khachiyan (1979) fut le premier à proposer un algorithme polynomial, c'est-à-dire dont le nombre d'itération grandit d'une manière polynomiale avec la taille du problème.

Malheureusement, cet algorithme s'avère inefficace en pratique. Il faudra attendre les travaux de Karmarkar (1984) [15] pour susciter engouement pour les méthodes de points intérieurs.

Actuellement, l'importance de ces méthodes a dépassé le cadre de l'optimisation linéaire et elles sont beaucoup utilisées dans le cadre de l'optimisation convexe, grâce notamment aux travaux de Nestrov et Nemiroveski (1994). L'idée de base de ces méthodes est d'utiliser des fonctions barrières (fonction de pénalité [13], fonction de potentiel [15], fonctions des centres [14], $\cdots$).

Les méthodes de points intérieurs sont devenues compétitives à la méthode du simplexe surtout pour les problèmes de grande taille (> 1000 variables ou contraintes). On distingue trois classes fondamentales de méthodes de points intérieurs à savoir

- Les méthodes affines.

- Les méthodes de réduction du potentiel.

- Les méthodes de trajectoire centrale.

2.3.1 Méthodes affines

L'idée remonte à Dikin (1967). Puis reprise et développée par plusieurs chercheurs au milieu des années 80. Il s'agit pratiquement de l'algorithme de Karmarkar sans fonction potentiel, et où à la place de la transformation projective on a une transformation affine. On présente la méthode affine primale,

Considèrons le programme linéaire primale suivant

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

Avec la matrice A de plein rang. On note S^0 l'intérieur de S, c'est-à-dire

$$S^0 = \{x \in \mathbb{R}^n : \ Ax = b, \, x > 0\}$$

On suppose que l'ensemble S^0 n'est pas vide.

Calcul de la direction de descente

Soit x_c le point courant tel que

$$\begin{cases} Ax_c = b \\ x_c > 0. \end{cases}$$

On cherche

$$x^+ = x_c + \alpha \triangle x$$

tel que

$$\begin{cases} c^t x^+ \leq c^t x_c \\ Ax^+ = b,\ x^+ > 0. \end{cases}$$

La direction de déplacement doit donc vérifier

$$\begin{cases} c^t \triangle x \leq 0 \\ Ax^+ = A(x_c + \alpha \triangle x) = b. \end{cases}$$

Sous les hypothèses que $\alpha > 0$, $\triangle x$ doit être dans le noyau de A, c'est-à-dire

$$\triangle x \in N(A) = \{x \in \mathbb{R}^n :\ Ax = 0\}.$$

La direction de plus forte descente est donnée par

$$\begin{cases} \min c^t \triangle x \\ sc \\ A\triangle x = 0 \\ \parallel \triangle x \parallel = 1 \end{cases}$$

dont la solution est

$$\triangle x = \frac{Proj_A(-c)}{\parallel Proj_A(-c) \parallel},$$

où $Proj_A$ étant une matrice orthogonale de projection sur le noyau de A. Comme A est de plein rang et en obtient la normalisation,

on a

$$\begin{aligned} \triangle x &= Proj_A(-c) \\ &= -(I - A^t(AA^t)^{-1}A)c. \end{aligned}$$

Remarque 2.3.1. *Une matrice orthogonale de projection P sur le noyau de A vérifie les propriètés suivantes*

$$\begin{aligned} AP &= 0 \\ P &= P^t \\ P^2 &= P. \end{aligned}$$

Il est alors facile de vérifier que

$$\begin{aligned} c^t \triangle x &= -c^t Proj_A(c) \\ &= -c^t Proj_A^2(c) \\ &= - \parallel Proj_A(c) \parallel^2 \leq 0. \end{aligned}$$

Longueur du pas

Puisque nous somme dans le cas linéaire, le taux de décroissance est constant dans la direction $\triangle x$. Le pas maximal est limité par les contraintes de non négativité. De plus, la transformation affine qui nous permettra de centrer le point n'est pas définie sur la frontière. Donc, il suffit de choisir

$$\alpha = \gamma \alpha_{max} \; avec \; 0 < \gamma < 1,$$

et

$$\alpha_{max} = \min_{\triangle x_i < 0} \frac{-(x_c)_i}{\triangle x_i}.$$

En pratique, on choisit $\gamma = 0.995$. Si $\triangle x \geq 0$ et $\triangle x \neq 0$, le problème est alors non borné.

Transformation affine

On fait une mise à l'échelle afin que le nouveau point x^+ soit loin de la frontière définie par $x > 0$. Un point idéal serait le vecteur unitaire e, ainsi, on cherche une transformation affine qui transforme x^+ en e. La matrice de transformation est simplement l'inverse de la matrice diagonale dont les composantes sont les mêmes que celles de x^+, cette matrice est inversible puisque $x^+ > 0$. Notons cette matrice par X.

On a alors

$$\begin{cases} X^{-1} x^+ = e \\ X^{-1} x = \overline{x} \end{cases}$$

Le programme linéaire devient (dans l'espace transformé)

$$\begin{cases} \min c^t X\overline{x} = \overline{c}^t\overline{x} \\ AX\overline{x} = \overline{A}\overline{x} = b \\ \overline{x} \geq 0. \end{cases}$$

Dans l'espace $-\overline{x}$ et étant donné le nouveau point $\overline{x_c} = e$, on obtient

$$\begin{aligned} \triangle\overline{x} &= Proj_{\overline{A}}(-\overline{c}) \\ &= -(I - \overline{A}^t(\overline{AA}^t)^{-1}\overline{A})\overline{c} \\ &= -(I - XA^t(AX^2A^t)^{-1}AX)Xc. \end{aligned}$$

Et

$$\overline{x}^+ = \overline{x}_c + \alpha\triangle\overline{x}$$

alors

$$\begin{aligned} x^+ &= X\overline{x}^+ \\ &= x_c + \alpha X\triangle\overline{x} \\ &= x_c - \alpha X(I - XA^t(AX^2A^t)^{-1}AX)Xc. \end{aligned}$$

Dans ce qui suit on résume l'agorithme de la méthode affine puis on donne les résultats de convergence.

2.3.2 Algorithme de la méthode affine primale

- **Initialisation** x_0 le point initial, $0 < \gamma < 1$, $\triangle c$ la décroissance de la fonction objective, ε le critère d'arrêt ;
- **Début**
  - $x_c = x_0$;
  - $\triangle c = \varepsilon(c^t x_c) + 1$;
- **Tant que** $\frac{\triangle c}{c^t x_c} > \varepsilon$ **faire**

- $X = X_c$
- $\triangle x = -X(I - XA^t(AX^2A^t)^{-1}AX)Xc$;
- $\alpha = \min\limits_{\triangle x_i<0} \frac{-(x_c)_i}{\triangle x_i}$
- $x^+ = x_c + \alpha\triangle x$
- $\triangle x = c^t x_c - c^t x^+$;
- $x_c = x^+$

– **Fin tant que**

– **Fin.**

Convergence de la méthode affine

On a le résultat suivant

Théorème 2.3.1. *[30] Sous les hypothèses que A est une matrice de plein rang, qu'il existe un point strictement intérieur et que la fonction objectif est non constante sur le domaine réalisable, alors*

1. *Si le problème primal et le problème dual sont non dégénérés, alors pour tout $\gamma < 1$, la suite générée par l'algorithme précédent converge vers une solution optimale.*
2. *Pour $\gamma \leq \frac{2}{3}$, la suite générée par l'algorithme précédent converge vers une solution optimale.*

Critère d'arrêt

Le critère d'arrêt devrait être relié à la satisfaction des conditions d'optimalité.

Considérons d'abord le cas non dégénéré (primal et dual) et soit le programme dual (DL)

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

Soit x_k la solution courante et considérons le programme

$$\begin{cases} \max \quad \| X_k s \| \\ s.c \\ A^t y + s = c \\ y \in \mathbb{R}^m, s \geq 0. \end{cases}$$

La solution de ce programme est donnée par

$$\begin{cases} y_k = (AX_k^2 A^t)^{-1} AX_k^2 c \\ s_k = c - A^t y_k. \end{cases}$$

On en déduit

$$\triangle x_k = -X_k^2 s_k.$$

Théorème 2.3.2. *Sous les hypothèses de non dégénérescence primales et duales, la solution* (y_k, s_k) *converge vers la solution optimale duale du problème* (DL) *lorsque* x_k *converge vers la solution optimale primale du problème* (PL).
Dans ce cas, un critère d'arrêt peut être défini sur la norme du vecteur de complémentarité.

Propriétés et commentaires de la méthode affine

1. C'est une méthode primale, performante en pratique, quoique sensible au choix initial.
2. Elle est simple au sens qu'elle ne demande ni transformation projective ni fonction potentiel, ni même la préparation du problème (forme canonique).
3. La démonstration de la convergence est relativement simple dans le cas dégénéré, elle est fastidieuse en cas d'échéant.
4. Il n y'a pas de résultat de polynomialité, on pense plutôt que l'algorithme est d'une complexité exponentielle.

Variantes et extensions

Il existe une version affine duale (opérant sur (DL)) est une autre primale-duale opérant simultanément sur (PL) et (DL) pour laquelle on établit la polynomialité dans un cas très spécial.

2.3.3 Méthode de réduction du potentiel

Ces méthodes sont le fruit d'une grande partie des études acharnées menées par plusieurs chercheurs vers la fin des années 80. Elle sont désignées pour trouver les solutions améliorantes du programme non linéaire suivant

$$\begin{cases} minf(x,y,s) = q\ln(c^t x - b^t y) - \sum_{j=1}^{n} \ln(x_j) \\ sc \\ Ax = b,\ x \geq 0 \\ A^t y + s = c,\ y \in \mathbb{R}^m, s \geq 0. \end{cases}$$

Où la fonction objectif f est appelée **fonction potentiel** et q son paramètre.
C'est ce type de fonction que Karmarkar introduit dans son article avec $q = 1$.
Notons que le premier terme (qu'on désire ramener à -∞) de cette fonction est q fois le logarithme du saut de dualité, le second terme est la fonction barrière logarithmique, introduite pour empêcher les solutions réalisables de s'approcher de la frontière du domaine réalisable. Ainsi, la fonction potentiel est un intermédiaire pour réduire progressivement le saut de dualité à zéro. De plus, sous les hypthèses simples concernant le programme (PL), on montre facilement que le saut de dualité est borné supérieurement par la valeur de la fonction potentiel : $c^t x - b^t y \leq c_1 e^{\frac{f(x,y,s)}{q}}$, où $c_1 > 0$ est une constante dépendante de (PL).
Supposons maintenant q fixé et soit (x, y, s) l'itéré en cours, on cherche alors le nouvel itéré (x^+, y^+, s^+) avec une décroissance convenable du potentiel f. Une telle réduction est garantie avec un taux $\delta > 0$ (à chaque itération) et alors le saut de dualité est réduit

d'une portion fixe en au plus $[\frac{q}{\delta}]$ itérations.

Ce raisonnement permet d'exhiber une borne supérieure pour le nombre total d'itérations nécessaire pour obtenir une solution optimale approchée avec une précision $\varepsilon > 0$, à partir d'un point réalisable initial (x^0, y^0, s^0). Cette borne est donnée par la formule

$$\frac{q}{\delta}\left[\ln\left(\frac{c^t x^0 - b^t y^0}{\varepsilon}\right) + c_2\right]$$

où c_2 est une constante qui dépend de (PL) et de (x^0, y^0, s^0).

Propriétés et commentaires

1. Loin d'être aussi simples que les méthodes affines.
2. Plus attractives pour au moins deux raisons

   $a/$ Leur complexité polynomiale.

   $b/$ Leur compatibilité avec les techniques de recherche linéaire sur le pas de déplacement, ce qui entraîne un comportement numérique révalisant.
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.

Variantes

Il existe plusieurs méthodes de réduction de potentiel de différents types (primales, duales, primales-duales), certains utilisent la fonction ci-dessus pour d'autre on ajoute le terme $(-\sum_{j=1}^{n} \ln(s_j))$, histoire de gagner quelques propriétés théoriques supplémentaires, par contre le volume calculatoire de l'itération augmente.

2.3.4 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 : une complexité polynômiale et une convergence superlinéaire.

Les algorithmes de trajectoire centrale restreintent les itérés à un voisinage de la trajectoire centrale, ce dernier est une courbe de points strictement réalisable (voir [32]) Ceci étant, il faut signaler tout de même que sur le plan technique les méthodes de trajectoire centrale présentent des inconvénients d'ordre algorithmique et numérique entre autre : le problème d'initialisation et le coût excessif de l'itération lié au choix de la direction et au calcul du pas de déplacement. De même, les résultats de convergence sont liés en grande partie au positionnement des itérés. En effet, si les itérés et principalement le point initial n'est pas proche de la courbe issue de la trajectoire centrale, aucun résultat de convergence de l'algorithme n'est garantie.

Pour remédier le problème d'initialisation, plusieurs chercheurs ont proposé deux variantes.

1. **Méthodes de trajectoire centrale non réalisable**

   Cette approche est développée comme une alternative qui élimine carrément la phase d'initialisation dans les méthodes de trajectoire centrale, on démarre l'algorithme d'un point non nécessairement réalisable et on tente de réaliser à la fois : faisabilité et optimalité.

   Le premier résultat théorique sur les méthodes de points intérieurs non réalisables de type primales-duales à été obtenu par Kojima et al [22], ils ont montré que leur algorithme est globalement convergent.

2. **Méthode de trajectoire centrale avec poids**

   Le problème des méthodes de trajectoire centrale est d'ordre pratique, puisque si le point réalisable initial n' est pas au voisinage de la trajectoire centrale, l'algorithme ne démarre pas. Les chercheurs ont proposé d'introduire un poids associé à la fonction barrière logarithmique qui apporte un soulagement considérable, ces méthodes sont appelées méthodes de trajectoire centrale avec poids. Voir [12, 17].

Il existe aussi d'autre méthodes de points intérieurs de type trajectoire centrale, pour résoudre les problèmes de la programmation linéaire et non linéaire et le problèmes de la programmation semi-définie. Ces méthodes sont basées sur l'introduction des fonctions appelées fonctions noyaux, voir à titre d'exemple [2, 3], elles déterminent une nouvelle classe de directions de Newton mais elles sont aussi considérées comme une proximité pour mesurer la qualité des points par rapport à la trajectoire centrale. Ces fonctions satisfont quelques propriétés qui conduisent à un model très important pour développer ces méthodes.

2.3.5 Méthodes de trajectoire centrale primales-duales

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.

Présentation de la méthode

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

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

où $f_\mu(x)$ est la fonction pénalisé définie par

$$f_\mu(x) = c^t x - \mu \sum_{i=1}^{n} \ln(x_i)$$

et μ est un paramètre barièrre strictement positif.

De même à tout problème dual (DL), on associe le problème pénalisé suivant

$$(DL)_\mu \begin{cases} \max f_\mu(s), \mu > 0 \\ s.c \\ A^t y + s = c \\ y \in \mathbb{R}^m, s > 0. \end{cases}$$

Lemme 2.3.3. *La fonction f_μ est strictement convexe.*

Preuve. Considérons le problème primal (PL), on a

$$f_\mu(x) = c^t x - \mu \sum_{j=1}^{n} \ln(x_j).$$

En effet, $f_\mu(x) \in C^\infty$ et nous avons en particulier

$$\begin{cases} \nabla f_\mu(x) = c - \mu X^{-1} e \\ \nabla^2 f_\mu(x) = \mu X^{-2} \end{cases}$$

où

$$X = diag(x_1, \ldots, x_n)$$

$$e = (1, \cdots, 1)^t \in \mathbb{R}^n$$

X est une matrice définie positive, car $x_j > 0$ et $\mu > 0$ d'où X^{-2} l'est aussi. Alors $\nabla^2 f_\mu(x)$ est une matrice définie positive. D'où le résultat. ■

Comme f_μ est strictement convexe et les contraintes sont linéaires, donc les conditions d'optimalités (KKT) sont nécessaires et suffisantes et elles s'écrivent comme suit

$$(CNS(PL)_\mu) \begin{cases} Ax = b, x > 0 \\ c - \mu X^{-1} e - A^t y = 0 \end{cases}$$

on pose $\mu X^{-1}e = s$, il devient

$$(CNS(PL)_\mu) \begin{cases} Ax = b, x > 0 \\ A^t y + s = c, s > 0 \\ XSe = \mu e \end{cases} \tag{2.1}$$

où

$$X^{-1} = diag(\frac{1}{x_i})$$

$$S = diag(s_1, \ldots, s_n)$$

$$e = (1, \ldots, 1)^t \in \mathbb{R}^n.$$

De même pour $(DL)_\mu$, on obtient

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

Pour chaque valeur de $\mu > 0$, on résoud le système $(CNS(PL)_\mu)$ par la méthode de Newton, on obtient ainsi une suite des solutions paramétrisées $(x(\mu), y(\mu), s(\mu))$ qui converge vers la solution du problème initial lorsque μ tend vers 0.
La trajectoire centrale est l'ensemble de toutes les solutions $(x(\mu), y(\mu), s(\mu))$ du système $(CNS(PL)_\mu)$, elle est définie par

$$T_c = \{(x(\mu), y(\mu), s(\mu)) : \ \mu > 0\}.$$

Facteur de centralité (de proximité)

Le facteur de centralité est défini par

$$\delta(x, y, s) = \| \frac{X(\mu)s(\mu)}{\mu} - e \|$$

$(x(\mu), y(\mu), s(\mu))$ est à proximité de la trajectoire centrale, s'il appartient à l'ensemble suivant

$$S_{cent}(\sigma) = \{(x, y, s) \in W : \| X(\mu)s(\mu) - \mu e \| \leq \sigma\mu,\ 0 < \sigma < 1\}.$$

Avec

$$W = \{(x, y, s) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+ :\ Ax = b,\ A^t y + s = c,\ x > 0,\ s > 0\}.$$

Calcul de la direction

Comme le système (2.1) est non linéaire, une résolution directe est généralement impossible, alors on cherche des solutions approximatives suivant la trajectoire centrale en formant une suite décroissante des sauts de dualité $\{\mu^k = \frac{x^k(\mu)s^k(\mu)}{n}\}$ qui tend vers 0 à la limite. Pour ce type de système, on utilise généralement la méthode de Newton.

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

$$F(x, y, s) = \begin{pmatrix} Ax - b \\ A^t y + s - c \\ XSe - \mu e \end{pmatrix} \tag{2.2}$$

Avec $x, s > 0$.

La direction de Newton $\Delta W = (\triangle x, \triangle y, \triangle s)$ est une solution du système linéaire

$$\Delta F(x, y, s) \begin{pmatrix} \triangle x \\ \triangle y \\ \triangle s \end{pmatrix} = -F(x, y, s) \tag{2.3}$$

telle que $\Delta F(x, y, s)$ est la matrice Jacobienne du système (2.2) au point (x, y, s) donnée par

$$\Delta F(x, y, s) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ S & 0 & X \end{pmatrix} \tag{2.4}$$

La résolution du système (2.4) est équivalente à celle du système linéaire suivant

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \triangle x \\ \triangle y \\ \triangle s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu e - XSe \end{pmatrix} \tag{2.5}$$

Ce qui donne

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

Donc le nouvel itéré s'écrit sous la forme

$$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha(\triangle x, \triangle y, \triangle s)$$

on remplace μ par $\overline{\mu}$ tel que $\overline{\mu} = \sigma\mu$, $0 < \sigma < 1$ et on réitère jusqu'à l'optimalité, c-à-d on obtient une valeur $\overline{\mu}$ suffisament proche de zéro.

Calcul du pas de déplacement

Le pas de déplacement $\alpha > 0$ 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.,

$$x^+ = x + \alpha\triangle x > 0 \tag{2.6}$$

$$s^+ = s + \alpha\triangle s > 0 \tag{2.7}$$

D'après (2.6), on a $\alpha\triangle x > -x$. Ce qui est équivalent à

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

tel que

$$I_0 = \{i \in 1, \ldots, n : \triangle x_i < 0\}$$

De même pour (2.7), on a $\alpha \triangle s > -s$. Ce qui est équivalent à

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

tel que

$$I_1 = \{i \in 1, \ldots, n : \triangle s_i < 0\}$$

Donc il suffit de prendre le pas de déplacement suivant

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

Algorithme de trajectoire centrale

Début algorithme

Données : Soit $\varepsilon > 0$ un paramètre de précision.

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

Tant que $(x^k)^t s^k > \varepsilon$ **faire**

1. Prendre $\mu^{k+1} = \mu^k \sigma$, $0 < \sigma < 1$.
2. Calculer

$$\begin{cases} \triangle x^k = (S^k)^{-1}(\mu^{k+1} e - X^k S^k e) - (S^k)^{-1} X^k (\mu^{k+1} e - X^k S^k e) \\ \triangle y^k = -(A(S^k)^{-1} X^k A^t)^{-1} A (S^k)^{-1} (\mu^{k+1} e - X^k S^k e) \\ \triangle s^k = -A^t \triangle y^k. \end{cases}$$

3. Calculer $\alpha = \rho \min(\alpha_1^k, \alpha_2^k)$, $0 < \rho < 1$, tel que

$$\alpha_1^k = \begin{cases} \min(\frac{-x_i^k}{(\triangle x^k)_i}) \ \ si \ \ I_0 \neq \emptyset, \\ +\infty \ \ si \ \ I_0 = \emptyset. \end{cases}$$

$$\alpha_2^k = \begin{cases} \min(\frac{-s_i^k}{(\triangle s^k)_i}) \ \ si \ \ I_1 \neq \emptyset, \\ +\infty \ \ si \ \ I_1 = \emptyset. \end{cases}$$

4. Prendre

   $x^{k+1} = x^k + \alpha^k \triangle x^k$

   $y^{k+1} = y^k + \alpha^k \triangle y^k$

   $s^{k+1} = s^k + \alpha^k \triangle s^k$

5. Poser $k = k + 1$

Fin tant que

Fin algorithme.

2.4 Résultats de convergence

Lemme 2.4.1. *[31] Soit* $(\triangle x, \triangle y, \triangle s)$ *la solution du système (2.5), alors on a* $(\triangle x)^k(\triangle s)^k = 0$ *et* $\mu^{k+1} = (1 - \alpha(1 - \sigma))\mu^k$*, où* $0 < \sigma < 1$ *et* $\mu^{k+1} = \frac{(x^k)^t s^k}{n}$.

Corollaire 2.4.2. *L'algorithme résoud les problèmes* (PL) *et* (DL) *après* $O(\sqrt{n}L)$ *itérations qui correspondent à* $O(n^{3,5}L)$ *opérations arithmétiques.* L *est une constante liée à la taille du problème* (PL).

Remarque 2.4.1. *En pratique, on prend souvent une certaine liberté pour le choix des valeurs du paramètre barrièrre, par exemple on peut prendre* $\mu^0 = \frac{((x^0)^t s^0}{n}$ *et* $\mu^{k+1} = \frac{\mu^k}{10}$.

Chapitre 3

Méthode barrière logarithmique via les fonctions majorantes

3.1 Introduction

On propose dans ce chapitre une approche barrière pour résoudre un problème de programmation linéaire. La méthode est basée sur une fonction de pénalisation logarithmique donnant lieu à un algorithme de type Newton dont l'efficacité est mise en valeur à travers des tests numériques moyennant une fonction majorante. Les fonctions majorantes servent pour approximer les fonctions barrières afin d'aboutir à un calcul simple et moins coûteux du pas de déplacement à chaque itération vis-à-vis des méthodes de recherche linéaire.

On s'intéresse à la résolution du problème suivant :

$$(DL)\begin{cases} \min b^t y \\ A^t y \geq c \\ y \in \mathbb{R}^m, \end{cases}$$

où

$A \in \mathbb{R}^{m\times n}$ tel que $rang A = m < n$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$.

Le problème (DL) peut s'écrire sous la forme standard suivante

$$(DL)\begin{cases} \min b^t y \\ A^t y - s = c \\ y \in \mathbb{R}^m,\ s \in \mathbb{R}^n \quad s \geq 0. \end{cases}$$

Son dual est écrit sous la forme

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

Notons par

$S_{DL} = \{y \in \mathbb{R}^m,\ A^t y - c \geq 0\}$, l'ensemble des solutions réalisables de (DL).

$S^0_{DL} = \{y \in \mathbb{R}^m,\ A^t y - c > 0\}$, l'ensemble des solutions strictement réalisables de (DL).

$S_{PL} = \{Ax = b, x \geq 0\}$, l'ensemble des solutions réalisables de (PL).

$S^0_{PL} = \{Ax = b, x > 0\}$, l'ensemble des solutions strictement réalisables de (PL).

Soient $u, v \in \mathbb{R}^n$, le produit scalaire associé est défini par

$$\langle u, v\rangle = u^t v = \sum_{i=1}^{n} u_i v_i.$$

On suppose aussi que S^0_{DL} et S^0_{PL} sont non vides.

Pour étudier théoriquement et numériquement le problème (DL), on le remplace par une suite de problèmes perturbés sans contraintes comme suit

$$(DL_r)\begin{cases} \min f_r(y) \\ y \in \mathbb{R}^m, \end{cases}$$

avec $r > 0$ est un paramètre barrière et f_r la fonction barrière définie par

$$f_r(y) = \begin{cases} b^t y + nr \ln r - r \sum_{i=1}^{n} \ln \langle e_i, A^t y - c\rangle & \text{si } \ A^t y - c > 0, \\ +\infty & \text{sinon.} \end{cases}$$

Pour résoudre le problème (DL_r), on utilise une méthode de points intérieurs moyennant les approches newtoniennes, et il faut s'assurer qu'à chaque itération que le nouvel itéré y_{k+1} vérifie la condition de réalisabilité. Pour cela, on introduit le pas de déplacement t_k dans l'itération de Newton qui s'écrit sous la forme $y_{k+1} = y_k + t_k d_k$ où d_k est la direction de descente. Pour calculer ce pas de déplacement, on peut utiliser une méthode de recherche linéaire telle que la méthode de Goldestein-Armijo, Fibonacci,... etc, malheuresement ces méthodes sont coûteuses en volume calculatoire.
Dans notre approche, on essaye d'éviter l'utilisation des recherches linéaires où on va approximer la fonction

$$\theta(t) = \frac{1}{r}\left[f_r\left(y + td\right) - f_r\left(y\right)\right]$$

par des fonctions majorantes dont le but de calculer le pas de déplacement d'une manière facile et beaucoup moins coûteuse que les méthodes de recherche linéaire.

Notre travail est réparti en deux parties ; la première partie concerne l'aspect théorique où on montre l'existence et l'unicité de la solution optimale du problème (DL_r) ainsi que sa convergence vers une solution optimale du problème (DL). La deuxième partie concerne l'aspect numérique où on présente l'algorithme de résolution de (DL) on donnant de nouvelles fonctions majorantes qui permettent de calculer explicitement le pas de déplacement. Cette étude est consolidée par des tests numériques présentés à la fin de ce chapitre.

3.2 Aspect théorique du problème (DL_r)

3.2.1 Existence de la solution du problème (DL_r)

Pour prouver que le problème (DL_r) possède une solution optimale, on montre que la fonction f_r est inf-compact, pour cela il suffit de prouver que son cône de recession

$$S_0((f_r)_\infty) = \{d \in \mathbb{R}^n, (f_r)_\infty(d) \leq 0\}$$

se reduit à l'origine c-à-d

$$(f_r)_\infty (d) \leq 0 \Longrightarrow d = 0,$$

où $(f_r)_\infty$ est définie par

$$(f_r)_\infty (d) = \lim_{t \to +\infty} \frac{f_r (y + td) - f_r (y)}{t} = b^t d.$$

Ce qui revient à prouver le lemme suivant

Lemme 3.2.1. *Supposons que S^0_{DL} et S^0_{PL} sont non vides, alors si $b^t d \leq 0$ et $A^t d > 0$ alors $d = 0$.*

Preuve.

Supposons que $d \neq 0$. Comme l'ensemble des solutions strictement réalisables du problème (PL) est non vide, alors il existe $x > 0$ tel que $Ax = b$.

On a : $\langle b, d \rangle = \langle Ax, d \rangle = \langle x, A^t d \rangle > 0 \Longrightarrow b^t d > 0$, contradiction.

D'ou f_r est inf-compact, donc le problème (DL_r) admet au moins une solution optimale.

■

3.2.2 Unicité de la solution du problème (DL_r)

On sait que si la matrice hessienne $H = \bigtriangledown^2 f_r (y)$ est définie positive, alors le problème (DL_r) est strictement convexe et s'il possède une solution optimale elle est unique.

On a

$$f_r (y) = b^t y + nr \ln r - r \sum_{i=1}^{n} \ln \left\langle e_i,\ A^t y - c \right\rangle,$$

d'où

$$\nabla f_r (y) = b - r \sum_{i=1}^{n} \frac{A e_i}{\langle e_i,\ A^t y - c \rangle},$$

et

$$\bigtriangledown^2 f_r (y) = r \sum_{i=1}^{n} \frac{A e_i (A e_i)^t}{\langle e_i\ ,\ A^t y - c \rangle^2}$$

les matrices $B_i = Ae_i (Ae_i)^t \in \mathbb{R}^{m\times m}, i = 1, ..., n$ sont définies positives. En effet $\forall y \in \mathbb{R}^m$ tel que $y \neq 0$ et pour tout $i = 1, ..., n$

$$\begin{aligned} \left\langle Ae_i (Ae_i)^t y,\ y \right\rangle &= \left\langle e_i e_i^t A^t y,\ A^t y \right\rangle \\ &= \left\langle e_i^t A^t y,\ e_i^t A^t y \right\rangle \\ &= \| e_i^t A^t y \|^2 . \end{aligned}$$

D'où B_i sont des matrices définies positives (car A est de plein rang), ce qui donne H définie positive.

Comme f_r est inf-compact et strictement convexe, le problème (DL_r) admet une solution optimale unique.

3.2.3 Convergence de (DL_r) vers (DL) lorsque r tend vers zéro

Pour tout $y \in S^0_{DL}$ et $r > 0$, on pose

$$\varphi(y, r) = f_r(y) = \begin{cases} f(y) + nr \ln r - r \sum_{i=1}^{n} \ln \langle e_i\ ,\ A^t y - c \rangle & \text{si } A^t y - c > 0 \\ +\infty & \text{sinon.} \end{cases}$$

Tel que $f(y) = b^t y$.

Lemme 3.2.2. *Pour $r > 0$, soit y_r une solution optimale du problème (DL_r), alors $y = \lim_{r\to 0} y_r$ est une solution optimale du problème (DL).*

Preuve.

La fonction φ est différentiable au point $(y_r,\ r)$ et on a

$$\nabla_y \varphi(y_r, r) = \nabla f_r(y_r) = 0.$$

On a $\forall y \in S^0_{DL}$

$$f(y) = \varphi(y, 0) \geq \varphi(y_r, r) + \langle y - y_r, \nabla_y \varphi(y_r, r) \rangle + (0 - r) \nabla_r \varphi(y_r, r).$$

Donc

$$f(y) \geq f(y_r) + nr\ln r - r\sum_{i=1}^{n}\ln\left\langle e_i\ ,\ A^t y_r - c\right\rangle - r\left[n\ln r + n - \sum_{i=1}^{n}\ln\left\langle e_i, A^t y_r - c\right\rangle\right]$$

ce qui donne

$$f(y) \geq f(y_r) - nr, \forall y \in S_{DL}^0$$

d'où

$$\min_{y\in S_{DL}} f(y) \geq f(y_r) - nr$$

et

$$f(y_r) \geq \min_{y\in S_{DL}} f(y) \geq f(y_r) - nr.$$

Si r tend vers zéro alors on a

$$\min_{y\in S_{DL}} f(y) = \lim_{r\to 0} f(y_r)$$

d'où si y_r est une solution optimale du problème (DL_r) alors il existe y^* une solution optimale du problème (DL) tel que

$$\lim_{r\to 0} y_r = y^*.$$

■

3.3 Aspect numérique du problème (DL_r)

Dans cette partie, nous nous intéressons au calcul numérique de la solution optimale du problème (DL_r) par des algorithmes efficaces. Depuis l'apparition de la fameuse méthode de Karmarkar en 1984, les méthodes de points intérieurs ont pris une attention considérable dans la résolution de différents problèmes d'optimisation et ceci grâce à leur comportement numérique polynomial. En effet, les méthodes de points intérieurs de type barrière logarithmique sont conçues pour résoudre ce type de problème en étant basées sur les conditions d'optimalité qui sont nécessaires et suffisantes.

En outre, y_r est une solution optimale du problème (DL_r) s'il satisfait la condition suivante

$$\bigtriangledown f_r(y_r) = 0. \tag{3.1}$$

Pour résoudre l'équation (3.1), nous utilisons l'approche de Newton qui trouve à chaque itération un vecteur $y_{rk} + d_k$, où d_k est un vecteur de déplacement solution du système linéaire suivant

$$H_k d_k = - \bigtriangledown f_r(y_{rk}). \tag{3.2}$$

Comme $H_k = \bigtriangledown^2 f_r\,(y_{rk})$ est une matrice symétrique et définie positive, la méthode de Cholesky et la méthode de gradient conjugué sont les plus commodes pour résoudre le système (3.2).

Ainsi, Pour assurer la convergence de l'algorithme considéré pour cette approche vers la solution optimale de problème (DL_r), il devrait être s'assurer que tous les itérés $y_{rk} + d_k$ restent strictement réalisables. Pour cela, nous introduisons à chaque équation un pas de déplacement t_k qui satisfait la condition

$$A^t(y_{rk} + t_k d_k) - c > 0.$$

Algorithme Prototype

Pour la simplicité de présentation, nous considérons dans la suite y_k au lieu de y_{rk} et y au lieu de y_r.

Début algorithme

Initialisation : y_0 strictement réalisable de (DL), $d_0 \in \mathbb{R}$, ε précision donnée, $k = 0$.

- **Tant que** $|b^t d_k| > \varepsilon$ **faire**
  - Calculer la direction d_k solution de système $H_k d_k = - \bigtriangledown f_r(y_k)$.
  - Calculer le pas de déplacement t_k.
  - Poser $y_{k+1} = y_k + t_k d_k$ et $k = k + 1$.
- **Fin tant que.**

Fin algorithme.

3.3.1 Calcul du pas de déplacement t_k

Les méthodes de recherche linéaires sont connues par leur utilisation pour calculer le pas de déplacement optimale t_k. Il s'agit de minimiser la fonction unidimenssionelle

$$\varphi(t) = \min_{t>0} f_r(y + td)$$

Les méthodes de type recherche linéaire les plus connues sont celles de Goldestein-Armijo, Fibonnacci, ..., etc. Malheureusement, elles sont coûteuses en volume calculatoire, voir même inapplicables pour les problèmes de programmation semi-définie (SDP).
Pour remédier cette difficulté, on exploite l'idée proposée par J.P. Crouzeix et B. Merikhi [8] qui approxime la fonction

$$\theta(t) = \frac{1}{r}\left(f_r(y + td) - f_r(y)\right)$$

par des fonctions majorantes simples donnant à chaque itération k, un pas de déplacement t_k explicite d'une manière facile et beaucoup moins coûteuse que les méthodes de recherche linéaire.

Remarque 3.3.1. *Pour que la fonction $\theta(t)$ soit bien définie, il faut que toutes les itérations soient strictement réalisables pour (DL), c-à-d, que si le vecteur $y \in S^0_{DL}$, il faut que $(y + td)$ reste dans S^0_{DL}. Ce qui revient à trouver $\widehat{t}$ tel que pour tout $t \in \left[0, \widehat{t}\right]$,*

$$y + td \in S^0_{DL}.$$

Lemme 3.3.1. *Pour tout $t \in \left[0, \widehat{t}\right]$, la fonction $\theta(t)$ est bien définie et s'écrit sous la forme suivante*

$$\theta(t) = t \times \left(\sum_{i=1}^{n} z_i - \|z\|^2\right) - \sum_{i=1}^{n} \ln(1 + tz_i)$$

où

$$\widehat{t} = \sup\{t, 1 + tz_i > 0\}$$

avec

$$z_i = \frac{\langle e_i,\ A^t d\rangle}{\langle e_i,\ A^t y - c\rangle}, \forall i = 1, ..., n.$$

Preuve.

Nous avons

$$\begin{aligned}\theta(t) &= \frac{1}{r}\left(f_r(y+td) - f_r(y)\right)\\ &= \frac{1}{r}\left[b^t(y+td) - r\sum_{i=1}^{n}\ln\left\langle e_i,\ A^t(y+td)-c\right\rangle - b^t y + r\sum_{i=1}^{n}\ln\left\langle e_i,\ A^t y - c\right\rangle\right]\\ &= \frac{1}{r}b^t d - \sum_{i=1}^{n}\ln(1+t\frac{\langle e_i,\ A^t d\rangle}{\langle e_i,\ A^t y - c\rangle})\end{aligned}$$

on a aussi

$$\nabla f_r(y) = b - r\sum_{i=1}^{n}\frac{Ae_i}{\langle e_i,\ A^t y - c\rangle}$$

d'où

$$d^t\nabla f_r(y) = d^t b - rd^t\sum_{i=1}^{n}\frac{Ae_i}{\langle e_i,\ A^t y - c\rangle}$$

ce qui donne

$$d^t b = d^t\nabla f_r(y) + rd^t\sum_{i=1}^{n}\frac{Ae_i}{\langle e_i,\ A^t y - c\rangle} \tag{3.3}$$

la direction d vérifie $\bigtriangledown^2 f_r(y)\,d = -\nabla f_r(y)$

d'où

$$d^t\bigtriangledown^2 f_r(y)\,d = -d^t\nabla f_r(y)\,. \tag{3.4}$$

D'après (3.3), l'équation (3.4) s'écrit

$$\begin{aligned}d^t b = &\quad -d^t\bigtriangledown^2 f_r(y)\,d + rd^t\sum_{i=1}^{n}\frac{Ae_i}{\langle e_i\ ,\ A^t y - c\rangle}\\ &= -d^t\bigtriangledown^2 f_r(y)\,d + r\sum_{i=1}^{n}\frac{\langle e_i\ ,\ A^t d\rangle}{\langle e_i\ ,\ A^t y - c\rangle}\end{aligned}$$

on a aussi

$$\begin{aligned}d^t\bigtriangledown^2 f_r(y)\,d \quad &= rd^t\sum_{i=1}^{n}\frac{Ae_i\,(Ae_i)^t}{\langle e_i\ ,\ A^t y - c\rangle^2}d \quad = r\sum_{i=1}^{n}\frac{d^t(Ae_i)\times(Ae_i)^t\,d}{\langle e_i\ ,\ A^t y - c\rangle^2}\\ &= r\sum_{i=1}^{n}\frac{\langle e_i\ ,\ A^t d\rangle\times\langle e_i\ ,\ A^t d\rangle}{\langle e_i\ \ ,\ A^t y - c\rangle^2} \quad = r\sum_{i=1}^{n}\frac{\langle e_i\ ,\ A^t d\rangle^2}{\langle e_i\ ,\ A^t y - c\rangle^2}\end{aligned}$$

si on pose

$$z_i = \frac{\langle e_i,\ A^t d\rangle}{\langle e_i,\ A^t y - c\rangle}$$

alors

$$d^t \bigtriangledown^2 f_r(y)\, d = r \sum_{i=1}^{n} z_i^2 = r \|z\|^2$$

d'où

$$d^t b = -r \|z\|^2 + r \sum_{i=1}^{n} z_i$$

donc

$$\theta(t) = t \times \left(\sum_{i=1}^{n} z_i - \|z\|^2 \right) - \sum_{i=1}^{n} \ln(1 + t z_i), \qquad t \in \left[0, \widehat{t}\right[$$

avec

$$\widehat{t} = \sup \left\{ \ t, 1 + t z_i > 0 \quad \right\}, \quad \|z\| = \left(\sum_{i=1}^{n} z_i^2 \right)^{1/2}.$$

■

3.3.2 Calcul des différentes valeurs de $\widehat{t}$

On suppose connue la moyenne $\overline{x}$ et l'écart type d'une série statistique de n variables réelles.

On utilise les résultats suivants [9]

$$\overline{x} - \sigma_x \sqrt{n-1} \leq \min_i x_i \ \leq \overline{x} - \frac{\sigma_x}{\sqrt{n-1}}$$
$$\overline{x} + \frac{\sigma_x}{\sqrt{n-1}} \leq \max_i x_i \leq \overline{x} + \sigma_x \sqrt{n-1},$$

où

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \ \ et \ \ \sigma_x^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \overline{x}^2.$$

On donne la première valeur de $\widehat{t}$

$$\widehat{t_1} = \sup \left\{ \ t \quad , 1 + t\beta_1 > 0 \ \right\} \text{ avec } \beta_1 = \overline{z} - \sigma_z \sqrt{n-1}$$

une autre valeur $\widehat{t_2}$ est dûe au fait que $|z_i| \leq \|z\|$ pour tout i.

$$\widehat{t_2} = \sup \left\{ \ t \quad , 1 + t\beta_2 > 0 \ \right\} \text{ avec } \beta_2 = -\|z\|.$$

Lemme 3.3.2.
La fonction $\theta(t)$ est bien définie sur $\left[0,\widehat{t_1}\right[$, $\left[0,\widehat{t_2}\right[$

Preuve. On a

$$\theta(t) = t \times \left(\sum_{i=1}^{n} z_i - \|z\|^2\right) - \sum_{i=1}^{n} \ln(1+tz_i), \qquad t \in \left[0,\widehat{t}\right[$$

avec

$$\widehat{t} = \sup\left\{ \; t, 1+tz_i > 0 \quad \right\}, \quad \|z\| = \left(\sum_{i=1}^{n} z_i^2\right)^{1/2}.$$

Si on pose $x_i = 1+tz_i$ d'où $\overline{x} = 1+t\overline{z}$ et $\sigma_x = t\sigma_z$

D'après l'inégalité précédente on a

$$1+t(\overline{z}-\sigma_z\sqrt{n-1}) \leq \min_i(1+tz_i) \leq 1+t(\overline{z}-\frac{\sigma_z}{\sqrt{n-1}})$$

D'où

$$0 < 1+t(\overline{z}-\sigma_z\sqrt{n-1} \leq 1+tz_i, \, \forall i$$

Donc

$$\widehat{t_1} = \sup\left\{ \; t \quad , 1+t\beta_1 > 0 \; \right\} \text{ avec } \beta_1 = \overline{z}-\sigma_z\sqrt{n-1}$$

On a aussi

$$|\, z_i \,| \leq \|\, z \,\|, \, \forall i$$

D'où

$$0 < 1-t \,\|\, z \,\| \leq 1+tz_i \leq 1+t \,\|\, z \,\|.$$

$$\widehat{t_2} = \sup\left\{ \; t \quad , 1+t\beta_2 > 0 \; \right\} \text{ avec } \beta_2 = -\|z\|.$$

■

3.4 Fonctions majorantes de θ

On cherche une fonction majorante $\widehat{\theta}$ de θ sur $\left[0,\ \widehat{t_i}\right[$, $i=1,2$, tel que

$$\widehat{\theta}(0) = 0, \ \|z\|^2 = \widehat{\theta}''(0) = -\widehat{\theta}'(0).$$

On suppose que $x_i > 0$ pour tout $i = 1, ..., n$, alors d'après [9] on a

$$n \ln\left(\overline{x} - \sigma_x \sqrt{n-1}\right) \leq A \leq \sum_{i=1}^{n} \ln x_i \leq B \leq n \ln \overline{x}, \tag{3.5}$$

avec

$$A = (n-1) \ln\left(\overline{x} + \frac{\sigma_x}{\sqrt{n-1}}\right) + \ln\left(\overline{x} - \sigma_x \sqrt{n-1}\right)$$

$$B = \ln\left(\overline{x} + \sigma_x \sqrt{n-1}\right) + (n-1) \ln\left(\overline{x} - \frac{\sigma_x}{\sqrt{n-1}}\right)$$

soit $x_i = 1 + t z_i$, alors $\overline{x} = 1 + t\overline{z}$ et $\sigma_x = t\sigma_z$

3.4.1 Première fonction majorante

D'après l'inégalité (3.5) on a

$$\sum_{i=1}^{n} \ln x_i \geq A,$$

d'où

$$\sum_{i=1}^{n} \ln x_i + t \|z\|^2 \geq t \|z\|^2 + A,$$

ce qui donne

$$nt\overline{z} - \left(\sum_{i=1}^{n} \ln x_i + t \|z\|^2\right) \leq nt\overline{z} - t \|z\|^2 - A.$$

Donc la première fonction majorante est définie par

$$\theta_0(t) = t\gamma_0 - (n-1) \ln(1 + t\ \alpha_0) - \ln(1 + t\ \beta_0)$$

tel que

$$\begin{cases} \gamma_0 &= n\overline{z} - \|z\|^2 \\ \alpha_0 &= \overline{z} + \frac{\sigma_z}{\sqrt{n-1}} \\ \beta_0 &= \overline{z} - \sigma_z \sqrt{n-1} = \beta_1. \end{cases}$$

La fonction majorante θ_0 est définie et convexe sur $\left[0, \widehat{t_1}\right[$ et on a

$$\theta(t) \leq \theta_0(t)$$

avec

$$\theta_0(0) = 0,\ \theta_0''(0) = -\theta_0'(0) = \|z\|^2.$$

3.4.2 Seconde fonction majorante

Une autre fonction majorante plus simple que θ_0 peut être extraite de l'inégalité du lemme suivant.

Lemme 3.4.1. *Pour tout $t \in \mathbb{R}$ et $z \in \mathbb{R}^n$ on a :*

$$t\sum_{i=1}^{n} z_i + t\|z\| - \sum_{i=1}^{n} \ln(1+tz_i) + \ln(1-t\|z\|) \leq 0.$$

Preuve. Posons

$$h(t) = t\sum_{i=1}^{n} z_i + t\|z\| - \sum_{i=1}^{n} \ln(1+tz_i) + \ln(1-t\|z\|),$$

Alors

$$\begin{aligned} h'(t) &= \sum_{i=1}^{n} z_i + \|z\| - \sum_{i=1}^{n} \frac{z_i}{(1+tz_i)} - \frac{\|z\|}{(1-t\|z\|)} \\ &= \sum_{i=1}^{n} z_i \left(1 - \frac{1}{1+tz_i}\right) + \|z\| \left(1 - \frac{1}{1-t\|z\|}\right) \\ &= \sum_{i=1}^{n} z_i \left(\frac{tz_i}{1+tz_i}\right) + \|z\| \left(\frac{-t\|z\|}{1-t\|z\|}\right) \\ &= t\sum_{i=1}^{n} \frac{z_i^2}{1+tz_i} - \frac{t\|z\|^2}{1-t\|z\|} \\ &= t\sum_{i=1}^{n} z_i^2 \left(\frac{1}{1+tz_i} - \frac{1}{1-t\|z\|}\right), \end{aligned}$$

on a $|z_i| \leq \|z\|$ pour tout i, ce qui donne $-\|z\| \leq z_i \leq \|z\|$

d'où

$$\frac{1}{1+t\|z\|} \leq \frac{1}{1+tz_i} \leq \frac{1}{1-t\|z\|}.$$

Ce qui implique que la fonction h est une fonction décroissante sur son domaine de définition D_h.

Comme $h(0) = 0$, alors on obtient

$$h(t) \leq 0, \ \forall t \in D_h$$

■

Du lemme précédent on peut construire la deuxième fonction majorante ainsi :

on a

$$t\sum_{i=1}^{n} z_i - \sum_{i=1}^{n} \ln\left(1+tz_i\right) \leq -t\left\|z\right\| - \ln\left(1-t\left\|z\right\|\right),$$

ce qui donne

$$t\sum_{i=1}^{n} z_i - t\left\|z\right\|^2 - \sum_{i=1}^{n} \ln\left(1+tz_i\right) \leq -t\left\|z\right\| - t\left\|z\right\|^2 - \ln\left(1-t\left\|z\right\|\right),$$

donc on obtient

$$\theta\left(t\right) \leq \theta_2\left(t\right)$$

tel que

$$\theta_2\left(t\right) = -t\left(\left\|z\right\| + \left\|z\right\|^2\right) - \ln\left(1-t\left\|z\right\|\right).$$

La fonction majorante θ_2 est définie et convexe sur $\left[0, \widehat{t_2}\right[$ et on a

$$\theta_2\left(0\right) = 0,\ \theta_2''\left(0\right) = -\theta_2'\left(0\right) = \left\|z\right\|^2.$$

3.4.3 Troisième fonction majorante

On peut considérer une fonction qui contienne seulement un logarithme et plus simple que la fonction θ_0.

On considère les fonctions suivantes

$$\widetilde{\theta}\left(t\right) = \widetilde{\gamma}t - \widetilde{\delta}\ln\left(1+\widetilde{\beta}t\right),\ t \in \left[0, \widetilde{t}\right[$$

avec

$$\widetilde{t} = \sup\left\{\ t,\ \ 1+t\widetilde{\beta} > 0\ \right\}$$

Nous pouvons aussi penser à une autre fonction majorante θ_1 , plus simple que θ_0 et qui nous permettent de trouver une relation entre θ_0 et θ_2, i.e.,

$$\theta_0\left(t\right) \leq \theta_1\left(t\right) \leq \theta_2\left(t\right).$$

Pour cela, on pose $\widetilde{\beta} = \beta_0 = \beta_1$, $\widetilde{\gamma} = \gamma_0 = \gamma_1$ et $\widetilde{\delta} = \delta_1 > 1$, ce qui revient par la condition suivante

$$\|z\|^2 = \widetilde{\delta}\widetilde{\beta}^2 = \widetilde{\delta}\widetilde{\beta} - \widetilde{\gamma},$$

et qui donne

$$\theta_1(t) = (n\overline{z} - \|z\|^2)t - \frac{\|z\|^2}{(\overline{z} - \sigma_z\sqrt{n-1})^2} \ln\left(1 + (\overline{z} - \sigma_z\sqrt{n-1})t\right)$$

Lemme 3.4.2. *On pose $t^* = \min(\widehat{t_1}, \widehat{t_2})$, donc on a*

$$\theta(t) \leq \theta_0(t) \leq \theta_1(t) \leq \theta_2(t) \leq +\infty \quad \forall t \in [0, t^*[$$

Preuve. Pour tout $t \in [0, t^*[$.

L'inégalité $\theta(t) \leq \theta_0(t)$ est une conséquence directe de (3.5).

Posons $f(t) = \theta_1(t) - \theta_0(t)$

Et comme $\beta_0 = \beta_1$ et $\alpha_0 \geq \beta_0$, on a pour tout $t \in [0, t^*[$

$$f''(t) = \frac{\widetilde{\delta}\beta_0^2 - \beta_0^2}{(1 + t\beta_0)^2} - \frac{(n-1)\alpha_0^2}{(1 + t\alpha_0)^2} = \frac{(n-1)\alpha_0^2}{(1 + t\beta_0)^2} - \frac{(n-1)\alpha_0^2}{(1 + t\alpha_0)^2} \geq 0$$

Puisque on a $\alpha_0 \geq \beta_0$ alors

$$(1 + t\alpha_0)^2 \geq (1 + t\beta_0)^2$$

ce qui donne

$$\frac{1}{(1 + t\beta_0)^2} \geq \frac{1}{(1 + t\alpha_0)^2}$$

d'où

$$f''(t) \geq 0.$$

Et comme

$$f(0) = f'(0) = 0 \text{ et } f''(t) \geq 0,$$

alors

$$f(t) \geq 0, \forall t \in [0, t^*[.$$

De même si on pose $g(t) = \theta_2(t) - \theta_1(t)$, alors

$$g(0) = g'(0) = 0 \text{ et } g''(t) = \frac{\beta_2^2}{(1+t\beta_2)^2} - \frac{\delta_1\beta_1^2}{(1+t\beta_1)^2}$$

Et comme

$$\|z\|^2 = \delta_1\beta_1^2 \text{ et } \beta_2 = -\|z\|,$$

alors

$$g''(t) = \|z\|^2 \left(\frac{1}{(1+t\beta_2)^2} - \frac{1}{(1+t\beta_1)^2}\right) \geq 0.$$

On a

$$\begin{aligned} \beta_1 - \beta_2 &= \overline{z} - \overline{z} - \sigma_z\sqrt{n-1} + \|z\| \\ &= \frac{2\overline{z}\|z\| + \frac{1}{n}\|z\|^2 + n\overline{z}}{\overline{z} + \overline{z} - \sigma_z\sqrt{n-1} + \|z\|^2} > 0, \end{aligned}$$

d'où $\beta_1 \geq \beta_2$. Ce qui donne

$$(1+t\beta_1)^2 > (1+t\beta_2)^2$$

donc

$$\frac{1}{(1+t\beta_2)^2} - \frac{1}{(1+t\beta_1)^2} \geq 0,$$

d'où

$$g(t) \geq 0, \ \forall t \in [0, t^*[\,.$$

Donc on obtient

$$\theta(t) \leq \theta_0(t) \leq \theta_1(t) \leq \theta_2(t) \leq +\infty \ \forall t \in [0, t^*[\,.$$

En on déduit que la fonction θ_0 atteint son minimum en un point

$$t_0 = b - \sqrt{b^2 - c}$$

où

$$b = \frac{1}{2}\left(\frac{n}{\gamma_0} - \frac{1}{\alpha_0} - \frac{1}{\beta_0}\right) \text{ et } c = -\frac{\|z\|^2}{\alpha_0\beta_0\gamma_0}.$$

En revanche, les minimums des deux fonctions θ_1 et θ_2 sont atteints au point

$$t_i = \frac{\delta_i}{\gamma_i} - \frac{1}{\beta_i},\ i = 1, 2.$$

■

Lemme 3.4.3. *Soint y_{k+1} et y_k deux solutions strictement réalisables du problème (DL_r), obtenues respectivement à l'itération $k+1$ et k, donc nous avons $f_r(y_{k+1}) < f_r(y_k)$.*

Preuve. On a

$$f_r(y_{k+1}) \simeq f_r(y_k) + \langle \bigtriangledown f_r(y_k), y_{k+1} - y_k \rangle$$

avec

$$y_{k+1} = y_k + t_k d_k$$

Donc

$$\begin{aligned} f_r(y_{k+1}) - f_r(y_k) &\simeq \langle \bigtriangledown f_r(y_k), t_k d_k \rangle = t_k \langle - \bigtriangledown^2 f_r(y_k) d_k, d_k \rangle \\ &\simeq -t_k \langle \bigtriangledown^2 f_r(y) d_k, d_k \rangle < 0. \end{aligned}$$

car f_r est strictement convexe. D'où

$$f_r(y_{k+1}) < f_r(y_k).$$

■

3.5 Tests numériques

Dans cette section, nous présentons des tests numériques, pour confirmer et consolider les performances numériques de nos approches. Les exemples testés sont pris de la littérature [7, 19, 20] et sont réalisés en langage MATLAB R2008b sur Coeure Intel i3-3217U CPU (1.80 GHz) avec 4.00 Go RAM.

La précision considérée dans les différents exemples est $\varepsilon \in [10^{-2},\ 10^{-6}]$.

On désigne par

S_0 : la stratégie qui utilise la fonction majorante θ_0.

S_1 : la stratégie qui utilise la fonction majorante θ_1.

S_2 : la stratégie qui utilise la fonction majorante θ_2.

Nbr : le nombre d'itérations nécessaire pour obtenir la solution optimale.

temps : le temps de calcul en secondes.

(m, n) : la dimension des exemples testés.

Nous notons que les matrices utilisées dans les tests numériques sont des matrices pleines.

3.5.1 Exemples à dimension fixe

Soient les programmes linéaires suivants

Exemple 3.5.1.

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

La solution strictement réalisable est $y^{(0)} = (-3, -1, -1)^t$

La solution optimale trouvée est $y^* = (-1, -2, 0)^t$ *après :*

13 itérations en utilisant la stratégie S_0

13 itérations en utilisant la stratégie S_1

18 itérations en utilisant la stratégie S_2

Exemple 3.5.2.

$$A = \begin{pmatrix} -2 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -0 & 1 \\ -0 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}, \; b = (0, 0, -1)^t, c = (-3, 1, -1, 0, 0, 0)^t$$

La solution strictement réalisable est $y^{(0)} = (-1, -1, -2)^t$

La solution optimale trouvée est $y^* = (-0.5, -0.0713, -0.5)^t$ *après :*

9 itérations en utilisant la stratégie S_0

9 itérations en utilisant la stratégie S_1

12 itérations en utilisant la stratégie S_2

Exemple 3.5.3.

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

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

La solution strictement réalisable est $y^{(0)} = (-\frac{1}{2}, -4, -1, -1, -1, -1)^t$

La solution optimale trouvée est $y^* = (-0.5, -1.5, 0, 0, -1.5, 0)^t$ *après :*

25 itérations en utilisant la stratégie S_0

25 itérations en utilisant la stratégie S_1

31 itérations en utilisant la stratégie S_2

Exemple 3.5.4.

$$A = \begin{pmatrix} -1 & -2 & -3 & -4 & -5 & -5 & -4 & -3 & -2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -6 & -7 & -8 & -9 & -10 & -5 & -2 & -8 & -3 & -1 & 0 & -1 & 0 & 0 & 0 \\ -11 & -12 & -13 & -14 & -15 & -6 & -7 & -80 & -90 & -10 & 0 & 0 & -1 & 0 & 0 \\ -1 & -10 & -20 & -30 & -40 & -50 & -60 & -80 & -90 & -10 & 0 & 0 & 0 & -1 & 0 \\ -3 & -9 & -27 & -60 & -45 & -60 & -75 & -8 & -9 & -46 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$b_i = -10^4, i = 1, ..., 5; c = (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, 0, 0, 0)^t$

La solution strictement réalisable est $y^{(0)} = (-1, -1, -1, -1, -1)^t$

La solution optimale trouvée est $y^* = (0, 0, -0.0888, 0, -0.0078)^t$ *après :*

42 itérations en utilisant la stratégie S_0

42 itérations en utilisant la stratégie S_1

71 itérations en utilisant la stratégie S_2

Exemple 3.5.5.

$$A = \begin{pmatrix} -518 & -31 & -39 & -59 & -97 & -49 & -22 & -86 \\ -96 & -64 & -14 & -11 & -36 & -12 & -49 & -84 \\ -20 & -28 & -14 & -31 & -59 & -72 & -99 & -40 \\ -76 & -165 & -22 & -20 & -22 & -43 & -61 & -80 \\ -89 & -80 & -76 & -46 & -75 & -31 & -954 & -69 \\ -34 & -49 & -85 & -69 & -49 & -75 & -328 & -40 \\ -30 & -14 & -12 & -46 & -68 & -96 & -62 & -188 \\ -45 & -30 & -74 & -73 & -70 & -61 & -31 & -36 \\ -88 & -74 & -51 & -36 & -60 & -32 & -52 & -20 \\ -73 & -39 & -71 & -74 & -474 & -61 & -28 & -12 \\ -16 & -23 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -55 & -753 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -51 & -393 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -27 & -393 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -33 & -733 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -16 & -293 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -49 & -823 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -35 & -833 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -51 & -223 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -20 & -133 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

$c_i = 1,\ b_i = -10000; i = 1, ...10$

La solution strictement réalisable est $y^{(0)} = (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1)^t$

La solution optimale trouvée est $y^* = (0, 0, 0, 0, 0, 0, -0.0051, 0, -0.0129, -0.0039)^t$ *après :*

77 itérations en utilisant la stratégie S_0

78 itérations en utilisant la stratégie S_1

115 itérations en utilisant la stratégie S_2

Le tableau suivant résume les résultats obtenus pour les différentes tailles.

Exemple (m,n)	S_0		S_1		S_2	
	Nbr	temps	Nbr	temps	Nbr	temps
Ex.3.5.1. $(3,5)$	13	0.0063	13	0.0063	18	0.094
Ex.3.5.2. $(3,6)$	9	0.063	9	0.032	12	0.046
Ex.3.5.3. $(6,12)$	25	0.0094	25	0.094	31	0.109
Ex.3.5.4. $(5,15)$	42	0.0093	42	0.125	71	0.188
Ex.3.5.5. $(10,20)$	77	0.141	78	0.406	115	0.532

3.5.2 Exemple à dimension variable

$$(DL)\begin{cases} \min \sum_{i=1}^{m} 2y_i \\ y_i - 1 \geq 0, i = 1, ..., m, \ n = 2m. \end{cases}$$

$y_0 = (1.5, 1.5, ..., 1.5)^t \in \mathbb{R}^m$ est un point strictement réalisable.

On pose $r = 0.4$, $\varepsilon = 10^{-6}$.

La solution optimale est $y^* = (1, 1, ..., 1)^t \in \mathbb{R}^m$.

Le tableau suivant résume les résultats obtenus pour les différentes tailles.

(m,n)	S_0		S_1		S_2	
	Nbr	temps	Nbr	temps	Nbr	temps
$(100,200)$	22	0.406	28	3.937	60	8.594
$(200,400)$	23	34.562	29	41	84	116.828
$(300,600)$	23	132.719	30	167.391	102	570.875
$(400,800)$	24	305.578	30	370.984	117	1453.625

Commentaires

Les tests numériques effectués montrent l'efficacité de nos trois approches et offrent dans chaque stratégie une solution optimale du problème (DL) en un temps raisonnable moyennant un nombre d'itérations acceptable. La stratégie S_0 est la plus commode du fait qu'elle exige moins d'itération par rapport au deux autres stratégies, de plus, elle nécessite un temps de calcul largement faible vis-à-vis des deux autres stratégies. Cela nous paraît tout à fait attendu, car théoriquement la stratégie S_0 utilise la fonction majorante θ_0 qui est la plus proche (meilleure approximante) de la fonction θ.

Conclusion

Dans ce travail, nous avons étudié un problème de programmation linéaire noté (DL). Nous associons à ce problème un problème perturbé, noté (DL_r).

Nous avons étudié l'existence et l'unicité de la solution optimale du problème (DL_r), et nous avons montré ensuite que la solution optimale de problème (DL_r) converge vers la solution optimale de problème (DL) quand r tend vers 0.

Comme le problème (DL_r) est strictement convexe, les conditions d'optimalité sont nécessaires et suffisantes. Pour cela, nous avons utilisé la méthode de Newton qui nous permet de calculer une bonne direction et déterminer un nouvel itéré.

Pour calculer le pas de déplacement, plusieurs méthodes ont été proposées par les chercheurs, parmi lesquelles, les méthodes de la recherche linéaire qui sont trés coûteuses. Pour éviter ce problème, nous avons proposé dans ce travail une nouvelle approche basée sur la notion des fonctions majorantes. En effet, trois fonctions majorantes sont proposées pour déterminer aisément le pas de déplacement.

Les simulations numériques que nous avons effectués confirment que les approches que nous avons proposées offrent réellement un pas de déplacement en un temps raisonnable et après peu d'itérations. De plus, notre algorithme converge vers la même solution optimale du problème linéaire en question, moyennant les trois stratégies proposées. La première stratégie parait la meilleure approche par rapport aux deux autres en terme de nombre d'itérations et de temps de calcul.

Bibliographie

[1] M. Achache, A weighted path-following method for the linear complementarity problem, Studia Universitatis Babes-Bolyai. Series Information, 48 (2004) 61–73.

[2] A. Asadi, C. Roos, Infeasible interior-point methods for linear optimization based on large neighborhood. J Optim Theory App 170 (2016) 562–590.

[3] Y. Q. Bai and C. Roos, A primal-dual interior point method based a new kernel function with linear growth rate, in Proceedings of the 9$^{\text{th}}$ Australian Optimization Day, Perth, Australia, (2002).

[4] Y.Q. Bai, M. El Ghami, C. Roos, A new efficient large-update primal-dual interior-point method based on a finite barrier. SIAM J. Optim. 13 (2003), 766–782.

[5] Y.Q. Bai, G. Lesaja, C. Roos, G.Q. Wang, M. El Ghami, A class of large-update and small-update primal-dual interior point algorithms for linear optimization. J. Optim. Theory Appl. 138 (2008), 341–359.

[6] D. Benterki, J.P. Crouzeix , B. Merikhi, A numerical feasible interior point method for linear semidefinite programs, RAIRO-Operation Research, 41 (2007) 49–59.

[7] A. Coulibaly, Méthode de point intérieurs en programmation linéaire, Thèse de Doctorat, Université Blaise Pascal, Clemont-Ferrand, 1994.

[8] J.P. Crouzeix, B. Merikhi, A logarithm barrier method for semidefinite programming, RAIRO-Operations Research, 42 (2008) 123–139.

[9] J.P. Crouzeix, A. Seeger, New bounds for the extreme values of a finite sample of real numbers, Journal of Mathematical Analysis and Applicatons, 197 (2008) 411–426.

[10] Z. Darvay, A weighted-path-following method for linear optimization, Studia Universitatis Babes-Boluai. Series Information, 47 (2002) 3–12.

[11] D. den Hertog, Interior point approach to linear quadratic and convex programming, Kluwer Academic Publishers, Dordrecht, 1994.

[12] E. Djeffel, Etude numérique d'une méthode de la trajectoire centrale avec poids pour la programmation convexe à contraintes linéaires, Mémoire de magister, Université Hadj Lakhdar, Batna, Algérie, 2009.

[13] A.V. Fiacco, G.P. McCormick, Nonlinear Programming : Sequential Unconstrained Minimization Techniques, John Wiley and Sons, New York, 1968.

[14] P. Huard, Resolution of mathematical programming : Sequential unconstraints by the method of centers, In Abadie, editor in Nonlinear Programming. Worth-Holland Amsterdam, 1967.

[15] N. Karmarkar, New polynomial-time algorithm for linear programming, In proceeding of the 16 th Annual ACM, Symposium on Weory of computing, (1984) 303–311.

[16] Z. Kebbiche, Mise en oeuvre d'une méthode de trajectoire centrale pour les problèmes complémentaires linéaires monotones, Thèse de magistère, Département de Mathématiques, Université Ferhat Abbas, Sétif-1, 1997.

[17] Z. Kebbiche, Etude et extensions d'algorithmes de points intérieurs pour la programmation non linéaire, thèse de doctorat, Université Ferhat Abbas, Sétif-1, 2007.

[18] Z. Kebbiche, A. Keraghel, A. Yassine, Extension of a projective interior point method for linearly constrained convex programming, Applied Mathematics and Computation, 193 (2007) 553–559.

[19] A. Keraghel, Etude adaptative et comparative des principales variantes de l'algorithme de Karmarkar, Thèse de Doctorat, Université Joseph Fourier, Grenoble, France, 1989.

[20] A. Keraghel, D. Benterki, Sur les performances de l'algorithme de Karmarkar pour la programmation linéaire : Revue Roumaine des sciences techniques mécaniques appliquées, 46 (2001) 87–96.

[21] M. Kojima, N. Megiddo, and S. Mizuno, A primal-dual infeasible interior point algorithm for linear programming, Math. Programming, 61 (1993), 263–280.

[22] H. Mansouri, C. Roos, Simplified O(nL) infeasible interior-point algorithm for linear optimization using full Newton steps, Optimization Methods and Software. 22 (2007) 519–530.

[23] H. Mansouri, M. Zangiabadi, An adaptive infeasible interior point algorithm with full-Newton step for linear optimization. Optimization 62 (2013), 285–297.

[24] S. Mizuno, M. J. Todd, and Y. Ye, On adaptive step primal-dual interior point algorithms for linear programming, Math. Oper. Res., 18 (1993) 964–981.

[25] R. Naseri, A. Valinejad, An extended variant of Karmarkar's interior point algorithm, applied Mathematics and Computation Elsevier 184 (2007) 737–742.

[26] Y.E. Nestrov, A. Nemiroveski, Interior-point polynomial algorithms in convex programming, SIAM, 1994.

[27] C. Roos. A full-Newton step O(n) infeasible interior-point method for linear optimization. SIAM Journal on optimization, 16 (2006) 1110-1136.

[28] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, New Jersey, 1970.

[29] C. Roos, T. Terlaky, J.P. Vial, Theory and algorithms for linear optimization : An interior point approach, W. John & Sons, 1997.

[30] R. Saigal, Linear programming ; A Modern Integrated Analysis, International Seriers in Operations Research and Management Science, vol.1, Kluwer Academic Publishers, Novembre 1995.

[31] I. Touil, Etude comparative des performances d'une méthode de points interieurs de type trajactoire centrale pour la programmation semi-définie linéaire. Thèse de Magister, Université Ferhat Abbas Sétif-1, 2005.

[32] S. Wright, Primal-dual interior point methods, SIAM, Philadelphia, USA, 1997.

[33] Y. Ye, Interior point algorithms : Theory and analysis, John wiley & Sons, New York, Wiley-Interscience Series in Discrete Mathematics Optimization, 1997.

[34] Y. Ye, E. Tse, An extension of Karmarkar's projective algorithm for convex quadratic programming, Mathematical programming, 44 (1989) 157–179.

ملخص :

في هذه المذكرة نعرض دراسة نظرية، خوارزمياتية و عددية لطريقة حاجز لوغاريتمي لحل مسائل البرامج الخطية (PL). نهتم بحساب اتجاه الانتقال باستعمال طريقة نيوتن وخطوة الانتقال باستعمال توابع محدودة جديدة بهدف خفض تكلفة الحساب العددي. عندئذ عرضنا النتائج النظرية التي تضمن وجود ووحدانية الحل الأمثل للمسألة التقريبية لـ (PL) وكذا تقاربه إلى الحل الأمثل للمسألة (PL). في الاخير عززنا هذا العمل بواسطة تجارب عددية على الخوارزمية الناتجة عن ذلك.

كلمات مفتاحية : البرمجة الخطية، طريقة النقط الداخلية، طريقة الحاجز اللوغاريتمي، الدالة المحدودة.

Abstract:

This work concerns the theoretical, algorithmical and numerical study of logarithmic barrier method to solve a linear programming problem (LP). We focus on the computation of the displacement direction through Newton's approach, and the displacement step using new majorant functions in order to reduce the computational cost. Theoretical results are presented giving rise to the existence and uniqueness of the optimal solution of the approximate problem (LP) and its convergence to that of (LP). This work is consolidated by numerical tests on the resulting algorithm.

Keywords: Linear Programming, Interior Point Methods, Logarithmic Barrier methods, Majorant function.

Résumé :

Ce travail concerne l'étude théorique, algorithmique et numérique d'une méthode barrière logarithmique pour résoudre un problème de programmation linéaire (PL). Nous mettons l'accent sur le calcul de la direction moyennant l'approche de Newton, et le calcul du pas de déplacement en utilisant de nouvelles fonctions majorantes dans le but de réduire le coût de calcul. Des résultats théoriques sont présentés donnant lieu à l'existence et l'unicité de la solution optimale du problème approché de (PL) ainsi que sa convergence vers celle de (PL). Ce travail est consolidé par des tests numériques réalisés sur l'algorithme obtenu.

Mots-clés : Programmation linéaire, Méthode de Points Intérieurs, Méthode barrière logarithmique, Fonction majorante.