

**UNE SYNTHÈSE SUR LES MÉTHODES DU POINT
INTÉRIEUR**

par

EMNA KALLEL

mémoire présenté au Département de mathématiques et d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

**FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE**

Sherbrooke, Québec, Canada, janvier 1998



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-35688-4

Le 19 janvier 1998, le jury suivant a accepté ce mémoire, dans sa version finale.

Président-rapporteur: M. François Dubeau _____
Département de mathématiques et d'informatique

Membre: M. Abdelhamid Benchakroun _____
Département de mathématiques et d'informatique

Membre: M. Jean-Pierre Dussault _____
Département de mathématiques et d'informatique

*À mes parents,
en témoignage de mon amour
et de ma reconnaissance.*

*À mon mari, à mes frères,
à toute ma famille,
avec toute mon affection.*

SOMMAIRE

En 1984, Karmarkar a publié un algorithme de type point intérieur pour la programmation linéaire. Il a affirmé qu'en plus d'être de complexité polynomiale, il est plus efficace que le simplexe surtout pour des problèmes de grande taille.

Ainsi une recherche s'est déclenchée dans les méthodes de point intérieur et a donné comme résultat une grande variété d'algorithmes de ce type qui peuvent être classés en quatre catégories de méthodes : méthodes projectives, méthodes affines, méthodes du potentiel et méthodes de la trajectoire centrale.

Dans ce travail, nous allons présenter une synthèse de ces méthodes incluant les derniers développements dans ce domaine.

REMERCIEMENTS

Je tiens à exprimer tout d'abord ma profonde reconnaissance à mon directeur de recherche, le Docteur Abdelhamid Benchakroun, qui m'a toujours fait confiance et qui a su m'encourager à surmonter les difficultés tant professionnelles que personnelles. Je tiens aussi à le remercier pour son appui scientifique et financier et pour sa grande disponibilité qui m'a été essentielle pour la progression de ma recherche.

Je remercie également les membres de jury pour leurs commentaires sur ce travail.

Qu'il me soit permis d'exprimer ma gratitude à mes parents, Mahmoud Kallel et Essia Triki, pour leurs encouragements, leur patience et le soutien qu'ils m'ont apporté en toutes circonstances durant mes études.

Que mon mari , Zied Chtourou, trouve ici l'expression de mes sentiments, les meilleurs, pour sa compréhension et ses encouragements durant ce travail.

Enfin, je tiens à remercier tous mes amis au Département de mathématiques et informatique et ailleurs, pour leur disponibilité et leur soutien.

TABLE DES MATIÈRES

SOMMAIRE	iii
REMERCIEMENTS	iv
TABLES DES MATIÈRES	v
INTRODUCTION	1
CHAPITRE 1 — GÉNÉRALITÉS	3
1.1 Définitions et propriétés	3
1.2 Méthode de Newton pour un problème avec contraintes d'égalité	5
1.3 Complexité d'un programme linéaire	7
1.3.1 Taille du problème	7
1.3.2 Complexité	9

1.4	Purification	10
1.5	Initialisation	11
CHAPITRE 2 — LES MÉTHODES AFFINES		12
2.1	Introduction	12
2.2	Méthode affine primale	13
2.2.1	Principe de ces méthodes	13
2.2.2	Détermination du point x^{k+1}	14
2.2.3	Algorithme affine primal	16
2.3	Variante de l'algorithme primal	18
2.3.1	Remarque	21
2.4	Méthode affine duale	21
2.4.1	Algorithme affine dual	24
2.5	Convergence de la méthode	25
CHAPITRE 3 — LES MÉTHODES PROJECTIVES		31
3.1	Introduction	31
3.2	Principe général de l'algorithme de Karmarkar	32

3.2.1	Principe de la méthode	33
3.2.2	Détermination du point x^{k+1}	36
3.2.3	L'algorithme de Karmarkar	37
3.2.4	L'approche de Todd et Burrell : Dualité	43
3.3	Mise en oeuvre et complexité	48
3.4	Transformation d'un problème général	49
CHAPITRE 4 — LES MÉTHODES DU POTENTIEL		52
4.1	Introduction	52
4.2	Principe général de la méthode	53
4.2.1	Les fonctions de potentiel	54
4.3	Méthode du potentiel primale	58
4.3.1	Algorithme potentiel primal	58
4.3.2	Complexité de l'algorithme	61
4.4	Méthode de potentiel duale	62
4.5	Méthode du potentiel primale-duale	62
4.5.1	Algorithme de potentiel primal-dual	63

4.5.2	Complexité de l'algorithme potentiel primal-dual	64
CHAPITRE 5 — LES MÉTHODES DE LA TRAJECTOIRE CENTRALE		73
5.1	Introduction	73
5.2	Préliminaires	74
5.3	Les méthodes primales duales	76
5.3.1	Principe	77
5.3.2	Algorithme primal-dual	84
5.3.3	Les méthodes prédicteurs-correcteurs	87
5.3.4	Algorithme prédicteur-correcteur:	89
5.4	Les méthodes primales	97
5.4.1	Principe	97
5.4.2	Algorithme à grand pas	102
5.4.3	Complexité de l'algorithme	103
CHAPITRE 6 — CONCLUSION		107

INTRODUCTION

En 1947, G.B. Dantzig a proposé la méthode de simplexe : une technique de résolution très efficace pour des problèmes de programmation linéaire.

Depuis ce temps là, la programmation linéaire a suscité un grand intérêt chez les chercheurs qui ont écrit des centaines de livres et ont publié des milliers d'articles sur le sujet. Bien que la complexité de cette méthode soit exponentielle, dans la pratique elle s'est montrée très efficace et d'une grande utilité, surtout dans les domaines de la planification et de l'organisation.

En 1979, le mathématicien soviétique L.G. Khachiyan a mis en oeuvre le premier algorithme polynomial pour la programmation linéaire : l'algorithme des ellipsoïdes. Toutefois, l'algorithme proposé s'est révélé complètement inefficace dans la pratique.

En 1984, N. Karmarkar a révolutionné le domaine de la programmation linéaire en mettant en oeuvre un algorithme polynomial basé sur les méthodes de pénalités intérieures. Celui-ci s'est avéré un compétiteur sérieux de l'algorithme du simplexe. Depuis, une recherche intense s'est déclenchée dans ce domaine et a donné comme résultat une grande

variété d'algorithmes de ce type qui peuvent être classés en quatre catégories :

1. méthodes projectives;
2. méthodes affines;
3. méthodes du potentiel;
4. méthodes de la trajectoire centrale.

Ce travail se veut une synthèse de ces différentes méthodes : nous définissons leurs principes, nous proposons quelques algorithmes et nous étudions la complexité de certains d'entre eux.

Ce mémoire est structuré comme suit :

- Dans le premier chapitre, nous rappelons les résultats fondamentaux en programmation linéaire et plus particulièrement ceux concernant la dualité, nous définissons aussi la notion de complexité;
- Dans le deuxième chapitre, nous étudions la méthode affine qui est considérée comme étant la plus simple des méthodes du point intérieur;
- Dans le troisième chapitre, nous exposons la méthode projective de Karmarkar;
- Dans le quatrième chapitre, nous traitons de la méthode de potentiel dont la complexité est meilleure que celles des deux précédentes;
- Dans le dernier chapitre, nous étudions la méthode de la trajectoire centrale qui est la plus populaire de ces méthodes.

CHAPITRE 1

GÉNÉRALITÉS

Dans ce chapitre, nous rappelons brièvement certaines propriétés d'un problème de programmation linéaire, nous décrivons ensuite la méthode de Newton pour résoudre un problème avec contraintes d'égalité et enfin nous rappelons certaines notions de la théorie de la complexité d'un programme linéaire.

1.1 Définitions et propriétés

Un problème de programmation linéaire s'écrit sous la forme :

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

où A est une matrice $(m \times n)$, c , $x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

Pour tout problème (P) , on définit son problème dual de la façon suivante :

$$(D) \begin{cases} \max & b^T y \\ & A^T y \leq c \end{cases}$$

ce qui est équivalent à :

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

où s désigne une variable d'écart. Les ensembles des solutions réalisables de (P) et (D) seront notés respectivement F_P et F_D . Ainsi, on a :

$$F_P = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$$

et

$$F_D = \{(y, s) \in \mathbb{R}^m \times \mathbb{R}^n, A^T y + s = c, s \geq 0\}.$$

Théorème de dualité forte

- Si un des problèmes (P) ou (D) possède une solution optimale finie, il en est de même pour l'autre et leurs valeurs optimales sont égales;
- Si un des problèmes (P) ou (D) est non borné, alors le domaine réalisable de l'autre est vide.

Marge duale

Soient x une solution réalisable pour (P) et (y, s) une solution réalisable pour (D) . On appelle marge duale la quantité suivante :

$$c^T x - b^T y = x^T c - x^T A^T y = x^T (c - A^T y) = x^T s.$$

Notons que l'optimalité de (P) et (D) est caractérisée par $x^T s = 0$.

1.2 Méthode de Newton pour un problème avec contraintes d'égalité

On considère le problème :

$$\begin{cases} \min f(x) \\ g(x) = 0 \\ x \in \mathbb{R}^n. \end{cases}$$

Les conditions d'optimalité de Karush-Khun-Tucker sont données par le système :

$$(S_{kt}) \begin{cases} \nabla f(x) + \lambda^T \nabla g(x) = 0 \\ g(x) = 0 \end{cases}$$

où λ est le multiplicateur de Lagrange associé à la contrainte $g(x) = 0$. Pour résoudre ce système, on peut utiliser la méthode de Newton qui consiste à partir d'un point (x^k, λ^k) , à linéariser (S_{kt}) au voisinage de x^k et à définir (x^{k+1}, λ^{k+1}) comme la solution du système linéarisé

$$(S_{ktl}) \begin{cases} \nabla f(x^k) + \sum_{i=1}^m \lambda_i^k \nabla g_i(x^k) + [\nabla^2 f(x^k) + \sum_{i=1}^m \lambda_i^k \nabla^2 g_i(x^k)](x^{k+1} - x^k) \\ + \sum_{i=1}^m (\lambda_i^{k+1} - \lambda_i^k) \nabla g_i(x^k) = 0 \\ g(x^k) + \nabla g(x^k)(x^{k+1} - x^k) = 0. \end{cases}$$

Le système (S_{ktl}) est équivalent à :

$$(S'_{ktl}) \begin{pmatrix} H^k & (J^k)^T \\ J^k & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) - (J^k)^T \lambda^k \\ -g(x^k) \end{pmatrix}$$

où

$$H^k = \nabla^2 f(x^k) + \sum_{i=1}^m \lambda_i^k \nabla^2 g_i(x^k)$$

et

$$J^k = [\nabla g_1(x^k), \nabla g_2(x^k), \dots, \nabla g_m(x^k)].$$

Après simplification, (S'_{k+l}) s'écrit :

$$(S_N) \begin{pmatrix} H^k & (J^k)^T \\ J^k & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ -g(x^k) \end{pmatrix}.$$

Si H^k est inversible, la solution de (S_N) est donnée par :

$$\begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} H^{-1} - H^{-1}J^T(JH^{-1}J^T)^{-1}JH^{-1} & H^{-1}J^T(JH^{-1}J^T)^{-1} \\ (JH^{-1}J^T)^{-1}JH^{-1} & -(JH^{-1}J^T)^{-1} \end{pmatrix} \begin{pmatrix} -\nabla f(x^k) \\ -g(x^k) \end{pmatrix}.$$

Notons qu'en posant $d^k = x^{k+1} - x^k$, (S_N) s'écrit :

$$\begin{cases} H^k d^k + (J^k)^T \lambda^{k+1} = -\nabla f(x^k) \\ J^k d^k = -g(x^k) \end{cases}$$

où d^k est solution du problème quadratique :

$$(Q) \begin{cases} \min \frac{1}{2} d^T H^k d + \nabla f^T(x^k) d \\ J^k d^k + g(x^k) = 0 \end{cases}$$

et λ^{k+1} est le vecteur dual optimal de (Q) .

Dans ce qui suit on présente une application qui sera utile dans le dernier chapitre.

Application au cas linéaire

On considère le problème :

$$\begin{cases} \min c^T x - \mu \sum_{i=1}^n \log x_i \\ Ax = b \\ x > 0 \end{cases}$$

où μ est un réel strictement positif. Les conditions de K.K.T de ce problème s'écrivent :

$$\begin{cases} c - \mu X^{-1} e + A^T \lambda = 0 \\ Ax = b. \end{cases}$$

D'après ce qui précède, la direction de Newton est déterminée par :

$$\begin{pmatrix} \mu X^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} d_N \\ -\lambda \end{pmatrix} = \begin{pmatrix} -c + \mu X^{-1} e \\ 0 \end{pmatrix}.$$

1.3 Complexité d'un programme linéaire

1.3.1 Taille du problème

Soit le problème :

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

où A est une matrice $(m \times n)$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$. On suppose que les données du problème sont des entiers. On définit les constantes l et L qui vont jouer un rôle important dans ce qui suit. Le nombre l est défini comme la longueur totale des données dans une représentation binaire qui est la sommation du nombre de bits utilisés pour l'entrée des données A , b et c .

Le déterminant d'une matrice est dominé par le produit des normes de ses colonnes et alors $|\det M| < 2^l$ pour n'importe quelle matrice carrée M de $[A, b]$. On définit $L = 2l + n + 1$. Le lemme suivant montre que 2^{-L} est un nombre très petit.

Lemme 1 *Soit v^* la valeur optimale de (P) . Si x est un point extrême de F_P , alors on a :*

1. *Pour tout $i = 1, \dots, n$, $x_i = 0$ ou $x_i > 2^{-L}$.*
2. *$c^T x = v^*$ ou $c^T x > v^* + 2^{-L}$.*
3. *Si F_P est borné, alors pour tout $x \in F_P$, on a, $c^T x \leq 2^L$.*

Preuve

1. Soit x un point extrême de l'ensemble F_P . Le système $Ax = b$ peut s'écrire $A_B x_B + A_N x_N = b$ où x_B est un vecteur défini par les composantes positives de x , et A_B , une sous-matrice de A avec les colonnes correspondantes. Puisque $x_N = 0$, alors, on a $A_B x_B = b$.

Si x est un point extrême alors les colonnes dans A_B sont linéairement indépendantes. Supposons que A_B est une matrice carrée (sinon on élimine les équations redondantes pour la rendre carrée). Alors, en utilisant la méthode de Cramer, la solution de ce système est donnée par :

$$x_i = \frac{\Delta_i}{\Delta} \quad \forall \quad i \in B$$

où $\Delta = \det A_B$ et Δ_i est le déterminant de la matrice obtenue en substituant la colonne A_i par b . Soit l_1 le nombre total de bits utilisés par $\{A, b\}$, alors, on a : Δ_i est un entier et $|\Delta| < 2^{l_1} < 2^L$ donc, $x_i = 0$ ou $x_i > 2^{-L}$;

2. Soit x^* un point extrême. D'après le processus ci-dessus, on a soit $x_i^* = 0$ ou $x_i^* = \frac{\Delta_i^*}{\Delta^*}$ avec Δ_i^* , un entier et $\Delta^* < 2^l$. Il s'en suit que $\Delta \Delta^* c^T(x - x^*)$ est un entier et alors soit que $c^T(x - x^*) = 0$ ou $c^T(x - x^*) \geq \frac{1}{\Delta \Delta^*} \geq 2^{-2l} > 2^{-L}$;

3. Soit l_c le nombre total de bits utilisés par c . Considérons premièrement un point extrême x comme précédemment et soit $i \in B$. Puisque $|\Delta_i| \leq 2^{l_1}$, $\Delta \neq 0$ est un entier et $x_i \leq 2^{l_1}$, il en résulte que $c^T x \leq n 2^{l_1} \leq 2^L$, et puisque $c_i x_i \leq 2^{l_1 + l_c} = 2^l$, $c^T x \leq n 2^l \leq 2^L$. Ceci est vrai pour tout point extrême de F_P . En particulier, si F_P est borné, alors n'importe quel point de F_P peut s'écrire comme combinaison convexe de ces points extrêmes i.e $\forall x \in F_P, x = \sum_{j \in J} \lambda_j x^j / \lambda_j \geq 0 \quad \forall j \in J, \sum_{j \in J} \lambda_j = 1$, x^j point extrême de F_P et alors $c^T x = \sum_{j \in J} \lambda_j c^T x^j \leq 2^L \sum_{j \in J} \lambda_j = 2^L$. D'où le résultat. \square

Remarque 1 *Ce lemme signifie que si on trouve un point extrême avec un coût plus petit que $v^* + 2^{-L}$, nécessairement ce point est une solution optimale du problème (P).*

1.3.2 Complexité

En 1972 Klee et Minty [34] ont démontré que l'algorithme du simplexe est de complexité exponentielle. En 1979 Khachiyan [33] a publié Le premier algorithme polynomial pour la programmation linéaire, sa complexité était de $O(n^4L)$ opérations arithmétiques avec n la dimension de l'espace et L la longueur des données définie précédemment. Cette borne de complexité a été réduite en 1984 par Karmarkar [31]. L'algorithme de Karmarkar résout le problème de programmation linéaire en $O(nL)$ itérations, chaque itération implique la résolution d'un système linéaire dans \mathbb{R}^n . La complexité de chaque itération est de $O(n^3)$ opérations mais cette borne a été réduite par Karmarkar à $O(n^{2.5})$ opérations en utilisant des techniques d'inversion. En 1986, Renegar [48] a décrit le premier algorithme de $O(\sqrt{n}L)$ itérations, cet algorithme avait la même complexité totale que celui de Karmarkar, c'est-à-dire $O(n^{3.5}L)$ opérations. Son approche était basée sur le suivi de la trajectoire centrale. En 1987 Gonzaga [24] et Vaidya [68] ont obtenu simultanément des algorithmes avec une complexité de $O(n^3L)$ opérations, le premier algorithme utilisait la fonction de barrière traditionnelle et le deuxième était une extension des résultats de Renegar. Depuis 1987, avec l'article de Kojima, Mizuno et Yoshise [37], la scène a été graduellement dominée par les algorithmes primals-duals. Ces algorithmes sont efficaces dans la pratique et peuvent s'étendre à d'autres types de problèmes (problèmes quadratiques, problèmes convexes, ...), leur complexité est de l'ordre $O(\sqrt{n}L)$ itérations, cette complexité a été obtenue en premier par Kojima et al. [36] et aussi par Monteiro et Adler[46]. Jusqu'à maintenant la meilleure borne de complexité obtenue pour des problèmes de programmation linéaire est de $O(n^3L)$ opérations arithmétiques [70][3].

1.4 Purification

En général, les algorithmes des méthodes du point intérieur s'arrêtent en un point \bar{x} proche d'un point extrême du domaine réalisable avec une valeur de l'objectif assez petite ($c^T \bar{x} < v^* + 2^{-L}$). Cependant, une fois \bar{x} trouvé, on peut déterminer la solution optimale x^* du problème (P) en utilisant la procédure de purification suivante : Si n contraintes sont actives en \bar{x} , alors \bar{x} est une solution de base et on a fini. Sinon il existe une direction d non nulle appartenant au noyau de l'espace défini par les contraintes actives. A partir de \bar{x} , on se déplace le long de la direction d si $c^T d < 0$ ou le long de la direction $-d$ sinon, et ce, jusqu'à ce qu'une nouvelle contrainte devienne active (cela peut se produire toujours car le domaine réalisable est borné). On continue ainsi jusqu'à ce que l'on obtienne une solution de base x^* .

Remarque 2 1. *Le processus de recherche d'un point extrême optimal implique au plus $n - m$ étapes, puisque on démarre avec m contraintes actives et linéairement indépendantes en \bar{x} (ce sont les contraintes $Ax = b$) et qu'à chaque étape, au moins une nouvelle contrainte active est identifiée.*

2. *x^* est obtenue à partir de \bar{x} en un temps polynômial.*

3. *x^* est bien une solution de base optimale pour le problème (P).*

En effet: Premièrement, remarquons d'abord qu'à chaque étape, la solution que l'on obtient correspond à une valeur de l'objectif inférieure à $v^ + 2^{-L}$ (car les directions choisies sont des directions de descente). Deuxièmement, si l'on suppose que les données du problème (P) sont des entiers (A , b et c) alors, en une solution de base x la valeur de l'objectif est égale à $c_B^T B^{-1}b$ où B est la base associée à la solution de base en question. Mais, $c_B^T B^{-1}b$ peut s'écrire $\frac{N}{D}$ où $N \geq 1$ et $D = \det | B |$, or $D < 2^L$ et si x n'est pas optimale alors obligatoirement on a $\frac{N}{D} \geq \frac{1}{D} \geq 2^{-L}$, par suite $c^T x^* < v^* + 2^{-L}$.*

1.5 Initialisation

Tous les algorithmes des méthodes du point intérieur qu'on va proposer exigent un point initial strictement réalisable. Pour trouver ce point, plusieurs méthodes ont été proposées, et parmi les plus utilisées on retrouve celles de Monteiro et Adler [46] et de Anstreicher et Bosch [3]. Dans ce paragraphe, on va présenter une autre plus simple pour la recherche du point initial. Soit alors le problème (P) , on introduit une variable artificielle λ , et on considère le problème augmenté suivant :

$$(P_\lambda) : \begin{cases} \min c^T x + M\lambda \\ Ax + (b - Ae)\lambda = b \\ x \geq 0 \\ \lambda \geq 0 \end{cases}$$

où M est une constante très grande. Notons que la taille de ce problème est de même ordre que celle de notre problème original (P) . Puisque $(e, 1)$ est une solution strictement réalisable à (P_λ) , donc, on peut appliquer nos algorithmes à ce problème augmenté (P_λ) . Remarquons que pour M suffisamment grand et si le problème original admet une solution optimale, alors on montre qu'une solution optimale (x^*, λ^*) de (P_λ) est telle que $\lambda^* = 0$ et x^* solution optimale de (P) .

CHAPITRE 2

LES MÉTHODES AFFINES

2.1 Introduction

Les méthodes affines ont été introduites en premier par Dikin [15] en 1967, mais il a fallu attendre l'apparition du célèbre article de Karmarkar [32] en 1984 pour qu'on puisse les découvrir à nouveau grâce à Barnes [7], Cavalier et Soyster [14], Vanderbei, Meketon et Freedman [69]. Comme l'approche originale de Karmarkar [32], les algorithmes affines sont des algorithmes de point intérieur, c'est-à-dire des algorithmes qui progressent tout en demeurant strictement à l'intérieur du domaine réalisable. La convergence de ces méthodes a été étudiée premièrement par Dikin (algorithme affine à petits pas) puis par Barnes [8], Vanderbei et *al.* [69] et Kortanek et Shi [38]. Il a été prouvé que les algorithmes convergent sous l'hypothèse de non dégénérescence. Récemment, Tsuchiya [65][66], Terlaky et Tsuchiya [58], Tsuchiya et Muramatsu [67] et Tseng et Luo [64] ont pu montrer que la méthode converge même dans le cas dégénéré. Cependant, aucune borne de polynomialité n'a été prouvée jusqu'à maintenant. Ces méthodes ont été mises

en oeuvre par plusieurs chercheurs. Citons parmi eux : Monna et Morton [45], Adler et al. [2][1]; ces derniers ont obtenu de bons résultats. En effet certains codes basés sur ces algorithmes sont maintenant disponibles. AT&T a développé un logiciel sous le nom commercial de KORBEX qui met en oeuvre des algorithmes affines de type primal, dual et primal-dual. Dans ce chapitre, on commence par la méthode affine primale. Ensuite, on donne l'algorithme primal ainsi que sa variante, puis on propose la méthode affine duale et enfin on étudie la convergence de la méthode ainsi que sa mise en oeuvre.

2.2 Méthode affine primale

2.2.1 Principe de ces méthodes

Considérons le problème de programmation linéaire suivant :

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

où A est une matrice $(m \times n)$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

On suppose que l'on dispose d'un point x^0 strictement réalisable de (P) i.e $x^0 > 0$ et $Ax^0 = b$.

Le principe des méthodes affines consiste à générer une suite $\{x^k\}$ de points strictement réalisables vérifiant : $c^T x^k < c^T x^{k-1} < \dots < c^T x^0$ et convergeant vers une solution optimale de (P).

2.2.2 Détermination du point x^{k+1}

L'idée de base d'une itération est la suivante : soit x^k le point obtenu à l'itération k (le point x^k est strictement réalisable). On considère la transformation affine : $x \xrightarrow{Ta} u = (X^k)^{-1}x$ où X^k est la matrice diagonale dont les composantes sont les x_i^k , $i = 1, \dots, n$. Alors x^k se transforme en $u = e = (1, \dots, 1)^T$ et le problème (P) se transforme en (\hat{P}) défini par :

$$(\hat{P}) \begin{cases} \min \hat{c}^T u \\ \hat{A}u = b \\ u \geq 0 \end{cases}$$

où $\hat{c} = X^k c$ et $\hat{A} = AX^k$. Le point courant pour (\hat{P}) étant $u = e$, on construit alors un nouveau point $\hat{u} = e + \bar{\alpha}\hat{d}$ où \hat{d} est une direction réalisable de descente pour (\hat{P}) et $\bar{\alpha}$, un pas de déplacement. x^{k+1} est alors obtenu en utilisant la transformation inverse : $\hat{u} \xrightarrow{(Ta)^{-1}} x^{k+1} = X^k \hat{u}$, qui s'écrit aussi, $x^{k+1} = X^k \hat{u} = X^k(e + \bar{\alpha}\hat{d}) = x^k + \bar{\alpha}X^k \hat{d} = x^k + \bar{\alpha}d^k$ où $d^k = X^k \hat{d}$.

Choix de la direction

La direction de descente généralement utilisée dans le nouvel espace est $\hat{d} = -\hat{c}_p$, où \hat{c}_p est la projection de \hat{c} sur le noyau de la matrice \hat{A} . Ainsi, on a :

$$\begin{aligned} \hat{d} &= -[I - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A}]\hat{c} \\ &= -[I - (AX^k)^T(AX^k(AX^k)^T)^{-1}AX^k]X^k c \\ &= -[I - X^k A^T(A(X^k)^2 A^T)^{-1}AX^k]X^k c \\ &= -X^k c + X^k A^T(A(X^k)^2 A^T)^{-1}A(X^k)^2 c. \end{aligned}$$

En posant

$$w^k = (A(X^k)^2 A^T)^{-1}A(X^k)^2 c, \quad (2.1)$$

\hat{d} s'écrit :

$$\hat{d} = -X^k(c - A^T w^k) \quad (2.2)$$

où w^k est la solution du système :

$$(A(X^k)^2 A^T)w = A(X^k)^2 c. \quad (2.3)$$

Choix du pas

Nous supposons que $\hat{d} \neq 0$ (car sinon la solution courante est optimale). Le pas $\bar{\alpha}$ est choisi le plus grand possible de façon à ce que le nouveau point \hat{u} demeure réalisable.

Ainsi,

$$\bar{\alpha} = \max\{\alpha : e + \alpha \hat{d} \geq 0\} = \begin{cases} +\infty & \text{si } \hat{d} \geq 0 \\ \min\{\frac{-1}{\hat{d}_j}; \hat{d}_j < 0\} & \text{sinon.} \end{cases}$$

Notons que si $\hat{d} > 0$ alors $\bar{\alpha} = \infty$, par conséquent le problème (\hat{P}) est non borné et par la suite le problème (P) est aussi non borné. D'autre part, afin d'assurer la stricte réalisabilité, on prend :

$$x^{k+1} = x^k + \gamma \bar{\alpha} d^k \quad \text{avec } \gamma \in (0, 1).$$

Proposition 1 *Pour tout $k \geq 0$, on a : $c^T x^k - c^T x^{k+1} > 0$.*

Preuve

On a :

$$\begin{aligned} c^T x^k - c^T x^{k+1} &= c^T x^k - c^T x^k - c^T \gamma \bar{\alpha} d^k \\ &= -\gamma \bar{\alpha} c^T d^k \end{aligned}$$

or

$$\begin{aligned}c^T d^k &= c^T (X^k) \hat{d} \\ &= \hat{c}^T \hat{d} \\ &= -\hat{c}^T \hat{c}_p \\ &= -\|\hat{c}_p\|^2 \\ &= -\|\hat{d}\|^2\end{aligned}$$

d'où :

$$c^T x^k - c^T x^{k+1} = \gamma \bar{\alpha} \|\hat{d}\|^2 > 0.$$

Ainsi, on obtient le résultat. \square

La version primale de l'algorithme affine est résumée dans l'algorithme 1 qui suit.

2.2.3 Algorithme affine primal

Algorithme 1

– Étape 1.

– Poser $k = 0$ et prendre $x^k > 0$ tel que $Ax^k = b$. (Une méthode de recherche de la solution initiale sera exposée ultérieurement);

– Choisir $\gamma \in (0, 1)$.

– Étape 2.

– Poser $X^k = \text{diag}(x_1^k, \dots, x_n^k)$.

- Étape 3.
 - Calculer la solution w^k de l'équation (2.3).
- Étape 4.
 - Calculer \hat{d} d'après l'équation (2.2);
 - Poser $d^k = X^k \hat{d}$.
- Étape 5.
 - Si $d^k > 0$, stop. Le problème (P) n'est pas borné.
 - Sinon: $\bar{\alpha} = \min\{\frac{-1}{\hat{d}_j}; \hat{d}_j < 0\}$ et $x^{k+1} = x^k + \gamma \bar{\alpha} d^k$.
- Étape 6.
 - Si le critère d'arrêt est satisfait, stop. Sinon $k = k + 1$ et aller à l'étape 2.

Plusieurs critères d'arrêt peuvent être considérés. Nous citons les plus utilisés. Les deux premiers critères s'appliquent lorsque la valeur de l'objectif est non nulle alors que le troisième critère s'applique lorsque cette valeur est nulle.

Critère 1: Ce critère est basé sur la variation relative de la valeur de la fonction objectif. Soit $\epsilon > 0$ un paramètre de tolérance, alors l'algorithme s'arrête lorsque : $\frac{c^T x^k - c^T x^{k+1}}{c^T x^k} < \epsilon$, avec une solution optimale $x^* = x^{k+1}$.

Critère 2: Celui-ci est basé sur l'utilisation de la marge duale. Todd et Burrell (1986) ont montré que le vecteur w^k obtenu à l'étape 3 de l'algorithme primal est une solution approximative du problème dual et que la suite $\{w^k\}$ engendrée par cet algorithme converge vers la solution optimale du dual. Ainsi, si $\epsilon > 0$ alors l'algorithme s'arrête lorsque : $|\frac{b^T w^k - c^T x^{k+1}}{c^T x^k}| < \epsilon$.

Critère 3: Soit $\epsilon > 0$, l'algorithme s'arrête lorsque : $\frac{c^T x^k - c^T x^{k+1}}{\max\{1, |c^T x^k|\}} < \epsilon$.

2.3 Variante de l'algorithme primal

Les différents algorithmes affines proposés pour résoudre le problème (P) diffèrent uniquement par le choix du pas. Par exemple, le choix de Vanderbei et al. [69] est semblable à celui de l'algorithme 1, tandis que Dikin [15][16] et Barnes [7] ont procédé de la façon présentée dans la variante ci-dessous.

Puisqu'il est difficile de minimiser la fonction objectif sur le domaine réalisable tout entier, on va optimiser sur des ellipsoïdes incluses dans F_P jusqu'à l'obtention de la solution optimale. On procède de la façon suivante : on commence avec un point $x^0 \in F_P$, on considère l'ellipsoïde S_0 centré en x^0 contenue dans F_P , on optimise la fonction objectif $c^T x$ sur S_0 , on obtient x^1 , on considère alors l'ellipsoïde S_1 centré en x^1 contenue dans F_P , on optimise de nouveau $c^T x$ sur S_1 , on obtient x^2 et ainsi de suite jusqu'à atteindre l'optimum.

Lemme 2 Soit $\beta \in (0, 1)$ un scalaire, $x > 0$ et $S_\beta = \{v \in \mathbb{R}^n \mid \sum_{i=1}^n \frac{(v_i - x_i)^2}{x_i^2} \leq \beta^2\}$
alors $v \in S_\beta \Rightarrow v > 0$.

Preuve

Soit $v \in S_\beta$, alors pour tout i , on a $(v_i - x_i)^2 \leq \beta^2 x_i^2 < x_i^2$ et donc, $|v_i - x_i| < x_i$. En particulier, $-v_i + x_i < x_i$, d'où $v_i > 0$. \square

Fixons un x strictement réalisable à (P) . La relation $v \in S_\beta$ s'écrit $\|X^{-1}(v - x)\| \leq \beta$ où X est une matrice diagonale dont les composantes sont les x_i et $\|\cdot\|$ est la norme

euclidienne. L'ensemble S_β est alors un ellipsoïde centré en x . Soit $S_0 = S_\beta \cap \{v/Av = b\}$ une section de l'ellipsoïde S_β , S_0 est à son tour un ellipsoïde contenu dans le domaine réalisable. Après, on remplace le problème (P) par la minimisation sur l'ellipsoïde S_0 :

$$(P_\beta) \begin{cases} \min c^T v \\ Av = b \\ \|X^{-1}(v - x)\| \leq \beta. \end{cases}$$

En posant $d = v - x$, le problème (P_β) est alors équivalent au problème suivant :

$$(P_d) : \begin{cases} \min c^T d \\ Ad = 0 \\ \|X^{-1}d\| \leq \beta. \end{cases}$$

La solution optimale de ce problème est donnée par la proposition suivante.

Proposition 2 *On suppose que A est de plein rang et que c n'est pas combinaison linéaire des colonnes de A . Soit x un vecteur strictement positif. Alors la solution optimale d^* du problème (P_d) est donnée par :*

$$d^* = -\beta \frac{X^2(c - A^T p)}{\|X(c - A^T p)\|}$$

où

$$p = (AX^2A^T)^{-1}AX^2c.$$

De plus, le vecteur $v = x + d^*$ appartient à F_P et

$$c^T v = c^T x - \beta \|X(c - A^T p)\| < c^T x. \quad (2.4)$$

Preuve

Voir [13]. \square

Remarques

- Si $x = x^k$ à l'itération k , on a $p = w^k$ et $(d^*)^k = -\beta \frac{(X^k)^2(c - A^T w^k)}{\|X^k(c - A^T w^k)\|} = \frac{\beta}{\|\hat{d}\|} d^k$.
Donc, si $(d^*)^k$ est positive, alors il en est de même pour d^k et dans ce cas, le problème (P) est non borné;

- Si de plus, on remplace γ par β et l'étape 5 de l'algorithme 1 par $x^{k+1} = x^k + d^*$, on obtient alors la variante de l'algorithme 1;
- Comme $\min\{\frac{-1}{\hat{d}_j} : \hat{d}_j < 0\} \geq \frac{1}{\|\hat{d}\|}$, on peut nommer l'algorithme 1: algorithme affine à grands pas et l'algorithme 2: algorithme affine à petits pas.

Algorithme 2

- Étape 1.
 - Poser $k = 0$ et prendre $x^k > 0$ tel que $Ax^k = b$;
 - Choisir $\beta \in (0, 1)$.
- Étape 2.
 - Poser $X^k = \text{diag}(x_1^k, \dots, x_n^k)$.
- Étape 3.
 - Calculer la solution w^k de l'équation (2.3).
- Étape 4.
 - Calculer \hat{d} d'après l'équation (2.2);
 - Calculer: $d^k = X^k \hat{d}$ et $d^* = \beta \frac{d^k}{\|\hat{d}\|}$.
- Étape 5.
 - Si $d^* \geq 0$ stop, le problème (P) n'est pas borné;
 - Sinon, $x^{k+1} = x^k + d^*$.
- Étape 6.
 - Si le critère d'arrêt est satisfait, stop. Sinon $k = k + 1$ et aller à l'étape 2.

Les critères d'arrêt sont les mêmes que pour l'algorithme 1.

2.3.1 Remarque

Malgré sa simplicité et son efficacité, l'algorithme affine primal présente un inconvénient majeur qui consiste à perdre la réalisabilité en calculant la direction d . En effet, on doit trouver une direction d vérifiant $Ad = 0$, or dans la pratique, on cherche d vérifiant $Ad = \xi$ où ξ est le vecteur des erreurs résiduelles. On aura donc $A(x + \gamma \bar{\alpha} d) = b + \gamma \bar{\alpha} \xi$ et par la suite, l'erreur a tendance à augmenter au fil des itérations aboutissant alors à une solution non réalisable. Pour remédier à cet inconvénient, une approche duale a été développée.

2.4 Méthode affine duale

Rappelons que le dual du problème (P) est :

$$(D) \begin{cases} \max b^T y \\ A^T y \leq c \end{cases}$$

et que $F_D = \{y \in \mathbb{R}^m, A^T y \leq c\}$ est son domaine réalisable. Soit y^0 un point à l'intérieur strict de F_D i.e $c - A^T y^0 > 0$. L'algorithme affine dual consiste à générer une suite de points $\{y^1, y^2, \dots, y^k, \dots\}$ vérifiant :

$$c - A^T y^k > 0$$

et

$$b^T y^{k+1} > b^T y^k$$

et convergeant vers la solution optimale de (D) . Voyons plus en détail la méthode. Le problème (D) s'écrit aussi :

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

où $s \in \mathbb{R}^n$ est une variable d'écart. Soient y^k et s^k les points courants à l'itération k et soit S^k la matrice diagonale dont les composantes sont les s_i^k $i = 1, \dots, n$. Par la transformation affine $s \xrightarrow{T_s} \hat{s} = (S^k)^{-1}s$ le problème (D) s'écrit (\hat{D}) :

$$(\hat{D}) \begin{cases} \max b^T y \\ A^T y + S^k \hat{s} = c \\ \hat{s} \geq 0. \end{cases}$$

Soit $F_{\hat{s}} = \{\hat{s} / \exists y \in F_D, A^T y + S^k \hat{s} = c\}$.

Proposition 3 *On a les relations suivantes entre les ensembles F_D et $F_{\hat{s}}$ directions d_y et $d_{\hat{s}}$:*

$$\hat{s} = (S^k)^{-1}(c - A^T y); \quad (2.5)$$

$$y = (A(S^k)^{-2} A^T)^{-1} A(S^k)^{-1} ((S^k)^{-1} c - \hat{s}). \quad (2.6)$$

Si $d_{\hat{s}} = \hat{s}_2 - \hat{s}_1$, $d_y = y_2 - y_1$, où (\hat{s}_i, y_i) sont reliés par (2.5) et (2.6) alors :

$$d_{\hat{s}} = -(S^k)^{-1} A^T d_y; \quad (2.7)$$

et

$$d_y = -(A(S^k)^{-2} A^T)^{-1} A(S^k)^{-1} d_{\hat{s}}. \quad (2.8)$$

Preuve

La relation (2.5) est triviale. La relation (2.6) se démontre de la manière suivante : On a : $A^T y + S^k \hat{s} = c$. Donc, $(S^k)^{-1} A^T y + \hat{s} = (S^k)^{-1} c$. Ou encore, $(S^k)^{-1} A^T y = (S^k)^{-1} c - \hat{s}$. Par suite $(A(S^k)^{-2} A^T) y = A(S^k)^{-1} ((S^k)^{-1} c - \hat{s})$. D'où, $y = (A(S^k)^{-2} A^T)^{-1} A(S^k)^{-1} ((S^k)^{-1} c - \hat{s})$. Les relations (2.7) et (2.8) se déduisent directement des relations (2.5) et (2.6). \square

Remarque 3 $d_{\hat{s}} \in \mathcal{R} ((S^k)^{-1} A^T)$ où $\mathcal{R} ((S^k)^{-1} A^T)$ désigne l'ensemble image associé à la matrice $(S^k)^{-1} A^T$.

Choix de la direction

La direction utilisée à chaque itération est la projection du gradient de la fonction objectif (en respectant les variables d'écart). Or, seules les variables d'écart sont changées par la transformation T_s . Le gradient de la fonction objectif est :

$$\nabla_{\hat{s}}(b^T y) = (\nabla_{\hat{s}} y)^T \nabla_y (b y) = -((A(S^k)^{-2} A^T)^{-1} A(S^k)^{-1})^T b = -(S^k)^{-1} A^T (A(S^k)^{-2} A^T)^{-1} b,$$

or $d_{\hat{s}} \in \mathcal{R}((S^k)^{-1} A^T)$ avec $\mathcal{R}((S^k)^{-1} A^T)$ désigne l'ensemble image associé à la matrice $(S^k)^{-1} A^T$, donc il n'est pas nécessaire de faire la projection et la direction dans $F_{\hat{s}}$ est alors :

$$d_{\hat{s}} = -(S^k)^{-1} A^T (A(S^k)^{-2} A^T)^{-1} b, \quad (2.9)$$

d'où :

$$d_y = (A(S^k)^{-2} A^T)^{-1} b. \quad (2.10)$$

Choix du pas

En appliquant la transformation inverse $(T_s)^{-1}$ à $d_{\hat{s}}$, on obtient la direction suivante :

$$d_s = -A^T (A(S^k)^{-2} A^T)^{-1} b = -A^T d_y$$

- Si $d_s < 0$, comme dans l'algorithme primal, on choisit le pas $\tilde{\alpha}$ le plus grand possible de façon que le nouveau point demeure réalisable i.e. $y^{k+1} = y^k + \tilde{\alpha} d_y \geq 0$, ou encore $s^{k+1} = s^k + \tilde{\alpha} d_s \geq 0$ ce qui donne $\tilde{\alpha} = \min\{\frac{-s_i^k}{(d_s)_i}; i = 1, \dots, n\}$;
- Si $d_s \geq 0$, alors le problème (D) est non borné et par la suite (P) est non réalisable (d'après le théorème de dualité forte).

2.4.1 Algorithme affine dual

Algorithme 3

– Étape 1.

– Poser $k = 0$ et prendre y^k tel que $c - A^t y^k \geq 0$;

– Choisir $\gamma \in (0, 1)$.

Pour $k = 0, 1, \dots$, répéter.

– Étape 2.

– Poser $s^k = c - A^T y^k$ et $S_k = \text{diag}(s_1^k, \dots, s_n^k)$.

– Étape 3.

– Calculer la solution d_y de l'équation (2.10).

– Étape 4.

– Calculer $d_s = -A^t d_y$, $d_{\hat{s}} = S_k^{-1} d_s$

– Étape 5.

– Si $d_s \geq 0$ stop, le problème (D) n'est pas borné et par la suite (P) est non réalisable;

– Sinon, $\tilde{\alpha} = \min\{\frac{-s_i^k}{(d_s)_i}; d_s < 0\}$, $s^{k+1} = s^k + \gamma \tilde{\alpha} d_s$ et $y^{k+1} = y^k + \gamma \tilde{\alpha} d_y$.

– Étape 6.

– Si le critère d'arrêt est satisfait, stop. Sinon $k = k+1$ et aller à l'étape 2.

Le critère d'arrêt pour cet algorithme est semblable à celui de l'algorithme affine primal : $\left| \frac{b^T y^{k+1} - b^T y^k}{b^T y^k} \right| < \epsilon$ si la valeur optimale est non nulle et $\left| \frac{b^T y^{k+1} - b^T y^k}{\max\{1, b^T y^k\}} \right| < \epsilon$ sinon (ϵ paramètre de tolérance assez petit donné d'avance).

2.5 Convergence de la méthode

La convergence de la méthode affine a intéressé plusieurs chercheurs, néanmoins, elle n'a été établie que récemment. Dans ce qui suit, nous allons l'établir pour l'algorithme 2. Sans perdre de généralité, on impose les hypothèses suivantes :

1. La matrice A est de rang maximal.
2. c n'est pas combinaison linéaire des lignes de A et ceci pour assurer que la fonction objectif $c^T x$ n'est pas constante sur le domaine réalisable.
3. Le problème (P) admet une solution optimale.
4. $\overset{\circ}{F}_P = \{x, Ax = b, x > 0\}$ est non vide.
5. Chaque solution de base réalisable au problème (P) est non dégénérée.
6. Le coût réduit de toutes les variables hors base de chaque solution de base réalisable au problème (P) est non nul (non dégénérescence duale).

Remarque 4 Les hypothèses 5 et 6 impliquent qu'à chaque solution de base réalisable du problème primal correspondent une solution de base réalisable non dégénérée. De plus

les hypothèses 3, 5 et 6 impliquent que chacun des deux problèmes (P) et (D) admet une solution optimale unique.

La convergence est résumée dans le théorème suivant :

Théorème 1 *On considère l'algorithme affine à petit pas. Si les hypothèses sont satisfaites et si $0 < \beta < 1$, alors x^k et w^k convergent respectivement vers les solutions optimales du problème primal et du problème dual.*

Pour prouver le théorème, on a besoin du lemme suivant :

Lemme 3 *Soit $F_P = \{x \in \mathbb{R}^n / Ax = b, x > 0\}$. Supposons que A possède m lignes linéairement indépendantes et que chaque solution de base réalisable est non dégénérée. Soit \bar{x} un élément de F_P et soit $N = \{i / \bar{x}_i > 0\}$ alors :*

1. *Il existe des indices $B(1), \dots, B(m) \in N$, pour lesquels les vecteurs $A_{B(1)}, \dots, A_{B(m)}$ sont linéairement indépendants et la solution de base correspondante est réalisable. En particulier $|N| \geq m$.*
2. *Si $|N| = m$, alors \bar{x} est une solution de base réalisable.*

Preuve

1. On considère un certain $\bar{x} \in F_P$ et soit $N = \{i / \bar{x}_i > 0\}$. Si les colonnes $A_i, i \in N$, sont linéairement dépendantes, alors il existe des coefficients $\lambda_i, i \in N$, non tous nuls, tels que

$$\sum_{i \in N} \lambda_i A_i = 0.$$

Notons qu'on a

$$\sum_{i \in N} A_i (\bar{x}_i + \theta \lambda_i) = b, \quad \forall \theta.$$

Choisissons θ tel que $\bar{x}_i + \theta\lambda_i \geq 0$ pour tout $i \in N$, et $\bar{x}_k + \theta\lambda_k = 0$ pour un certain $k \in N$. On obtient alors un élément de F_P qu'on note \hat{x} tel que $\{i/\hat{x}_i > 0\}$ soit un sous espace propre de N . On répète cette procédure autant de fois que nécessaire jusqu'à obtenir un certain $\tilde{x} \in F_P$, pour lequel $I = \{i/\tilde{x}_i > 0\}$ soit un sous espace de N , et les colonnes $A_i, i \in I$, soient linéairement indépendantes. Si $|I| < m$, on peut augmenter les colonnes $A_i, i \in I$, pour obtenir m colonnes linéairement indépendantes (une base). La solution de base réalisable correspondante est \tilde{x} . Cette solution est dégénérée, et on arrive alors à une contradiction. Donc $|I| = m$. Ceci implique que $|N| \geq m$. De plus, puisque $I \subset N$, les colonnes $A_i, i \in I$, sont m colonnes linéairement indépendantes associées aux composantes positives de \bar{x} .

2. En utilisant ce qui précède, les m vecteurs $A_i, i \in N$, sont linéairement indépendants et, alors, \bar{x} est une solution de base réalisable. \square

Preuve du théorème

Soit x^* une solution de base réalisable optimale du problème (P) . D'après les hypothèses 5 et 6, les coûts réduits de toutes les variables hors base sont positifs, cela implique que la solution optimale x^* du problème (P) est unique. En utilisant l'optimalité de x^* et la proposition (2) avec $x = x^k, v = x^{k+1}$, on obtient :

$$c^T x^* \leq c^T x^{k+1} = c^T x^k - \beta \|X^k(c - A^T w)\| < c^T x^k. \quad (2.11)$$

Donc, la suite $c^T x^k$ est décroissante et bornée inférieurement, donc elle converge. En particulier, $c^T x^{k+1} - c^T x^k$ tend vers 0 et par la suite $X^k(c - A^T w^k)$ tend vers 0. En utilisant la définition de X^k , on obtient :

$$\lim_{k \rightarrow \infty} x_i^k (c_i - A_i^T w^k) = 0 \quad i = 1, \dots, n. \quad (2.12)$$

Donc la complémentarité est satisfaite à la limite.

Montrons que la suite x^k est bornée. Considérons alors le problème :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n x_i \\ Ax = b \\ c^T x \leq c^T x^* \\ x \geq 0. \end{array} \right.$$

Ce problème a une valeur optimale finie, car x^* est la seule solution réalisable. Il en est de même pour le problème :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n x_i \\ Ax = b \\ c^T x \leq c^T x^0 \\ x \geq 0. \end{array} \right.$$

Ainsi, l'ensemble $\{x/Ax = b, x \geq 0, c^T x \leq c^T x^0\}$ est borné. Puisque $c^T x^k \leq c^T x^0$, alors la suite x^k est bornée.

La suite x^k est bornée; donc, d'après le théorème de Bolzano Weirstrass, il existe une suite k_j d'entiers positifs telle que x^{k_j} converge vers un certain \bar{x} . Il est facile de voir que \bar{x} est une solution réalisable. Soit $N = \{i/\bar{x}_i > 0\}$, d'après le lemme 3, l'ensemble N a au moins m éléments : $B(1), \dots, B(m) \in N$ pour lesquels les colonnes $A_{B(1)}, \dots, A_{B(m)}$ sont linéairement indépendantes et la solution de base correspondante \hat{x} est réalisable. Soit B , la matrice de base correspondante, et $c_B = (c_{B(1)}, \dots, c_{B(m)})$, en utilisant l'équation (2.12), $x_i^{k_j} (c_i - A_i^T w^{k_j})$ converge vers zéro quand $j \rightarrow \infty$, pour tout i . $\forall i = 1, \dots, m$, on a $B(1), \dots, B(m) \in N$ et $x_{B(i)}^{k_j}$ converge vers $\bar{x}_{B(i)}$, qui est positive car $B(i) \in N$. Il s'ensuit que $c_{B(i)} - A_{B(i)}^T w^{k_j}$ converge vers zéro. En notation matricielle, on a $c_B - B^T w^{k_j}$ converge vers zéro. Ainsi, w^{k_j} converge vers $(B^T)^{-1} c_B$. Par conséquent, pour tout i , $c_i - A_i^T w^{k_j}$ converge vers $c_i - A_i^T (B^T)^{-1} c_B$, qui est le coût réduit de la i ème variable de la solution de base \hat{x} . Par la supposition (6), on a $c_i - A_i^T (B^T)^{-1} c_B \neq 0$ pour tout $i \neq B(1), \dots, B(m)$. Donc, pour tout $i \neq B(1), \dots, B(m)$, $c_i - A_i^T w^{k_j}$ converge vers une valeur non nulle. En utilisant de nouveau l'équation (2.12), on en déduit que $\bar{x}_i = 0 \quad \forall i \neq B(1), \dots, B(m)$.

En particulier, $N = \{B(1), \dots, B(m)\}$, et \bar{x} est une solution de base associée à la matrice B .

On a montré que chaque suite $\{x^k\}$ a au moins un point limite et que chaque point limite doit être une solution de base réalisable. Vérifions maintenant que ce point limite est unique. Soit $\delta > 0$ de sorte que chaque variable de base en chaque solution de base est plus grande que δ , d'après les hypothèses de la non dégénérescence, les variables de base sont toujours positives, et alors, un tel δ existe. Soit $\epsilon = \frac{\delta}{3}$, puisque chaque point limite de suite x^k est une solution de base réalisable, il existe K tel que pour tout $k > K$, $\|x^k - x_b\| < \epsilon$, où x_b est une certaine solution de base réalisable. (Dans le cas contraire, on peut construire une suite x^{k_i} telle que $\|x^{k_i} - x_b\| > \epsilon$, cette suite doit avoir un point limite, mais aucune solution de base réalisable ne peut être un de ces points limite. On aboutit alors à une contradiction. On suppose qu'il existe $k \geq K$ et $\bar{x} \neq \hat{x}$, deux solutions de bases telles que $\|x^k - \bar{x}\| \leq \epsilon$, $\|x^{k+1} - \hat{x}\| < \epsilon$, que la i ème variable de \bar{x} est hors base (i.e $\bar{x}_i = 0$) et que la i ème variable de \hat{x} est de base (i.e $\hat{x}_i \geq \delta$), il s'ensuit que $x_i^k \leq \epsilon$ et $x_i^{k+1} \geq \delta - \epsilon = 2\epsilon$. D'autre part, l'équation (2.4) donne :

$$x_i^{k+1} = x_i^k \left(1 - \beta \frac{x_i^k \hat{d}_i}{\|d^k\|}\right) < x_i^k (1 + \beta) < 2x_i^k \leq 2\epsilon \leq x_i^{k+1}.$$

Ceci est une contradiction, elle stipule que si $k \leq K$ et si x^k est tel que $\|x^k - \bar{x}\| \leq \epsilon$; alors, c'est vrai pour x^{k+1} . On en déduit qu'aucune solution de base réalisable autre que \bar{x} ne peut être un point limite, et donc x^k converge vers \bar{x} et w^k , vers la solution de base duale associée. Il reste à montrer que \bar{x} est nécessairement optimale. Si \bar{x} ne l'est pas, alors il existe une certaine variable hors base x_i dont le coût est négatif, Or on a déjà montré que $c_i - A_i^T w^k$ converge vers le coût réduit de x_i par conséquent $c_i - A_i^T w^k$ deviendrait et demeurerait négatif éventuellement, et donc, x_i^k sera une suite strictement croissante. D'autre part, x_i^k converge vers \bar{x}_i qui est nulle (car x_i est une variable hors

base). Il en résulte une contradiction. \square

Théorème 2 *On considère l'algorithme affine à grands pas. Si les hypothèses 1 à 4 sont satisfaites et si $0 < \gamma < \frac{2}{3}$, alors x^k et w^k convergent respectivement vers les solutions optimales du problème primal et du problème dual.*

Preuve

Le théorème (2) se démontre de la même façon que le précédent. \square

Notons que pour le problème :

$$(P) \begin{cases} \min x_1 + x_2 + x_3 \\ x_1 + x_2 + x_3 + x_4 = 0 \\ x_1, x_2, x_3, x_4 \geq 0, \end{cases}$$

on montre que si $\gamma > \frac{2}{3}$, la suite des itérations w^k ne converge pas.

Remarques

1. Chaque itération de l'algorithme affine exige un nombre d'opérations de l'ordre de $O(n^3)$;
2. Si l'algorithme démarre très proche d'un point extrême, il va tendre à circuler tout le long des arêtes du domaine réalisable. Bien que cela n'a jamais été prouvé, on conjecture que cette situation menera à un temps d'exécution exponentiel, dans le pire cas. De plus, si le point courant est proche de la frontière du domaine réalisable, alors, les ellipsoïdes peuvent devenir très petits et l'algorithme prendrait alors des petits pas, alors que si le point courant est centré, les ellipsoïdes sont larges et l'algorithme progresse rapidement. En effet, il a été constaté que la valeur de la fonction objectif décroît rapidement aux premières itérations, mais ce taux de décroissance devient très petit à l'approche d'une solution optimale.

CHAPITRE 3

LES MÉTHODES PROJECTIVES

3.1 Introduction

Malgré sa complexité exponentielle, l'algorithme du Simplexe est resté longtemps l'algorithme de référence pour la programmation linéaire. La méthode est demeurée sans compétiteurs sérieux jusqu'en 1984 où un nouvel algorithme polynomial, l'algorithme des méthodes projectives a été proposé par Narendra Karmarkar [32] aux laboratoires de AT & T Bell aux États Unis. Mais la mise en place et les transformations nécessaires pour résoudre un problème standard de programmation linéaire par la méthode de Karmarkar sont demeurées secrètes jusqu'en 1985 où elles ont été publiées par Tomlin [63] et Shanno et Marsten [54] en particulier. Cet algorithme a été essentiellement différent de la méthode du simplexe par le fait que, dans cette dernière, on se déplace en suivant la frontière du domaine réalisable, alors que pour la méthode de Karmarkar, on progresse tout en restant strictement à l'intérieur du domaine réalisable, d'où la connotation "intérieur", et c'était grâce à lui qu'une recherche intense dans les méthodes du point intérieur

s'est déclenchée et a donné comme résultat une grande variété d'algorithmes de ce type. L'algorithme de Karmarkar avait un grand intérêt théorique puisqu'il est polynômial; en effet, il exige un $O(nL)$ itérations et chaque itération exige $O(n^3)$ opérations arithmétiques, d'où une complexité totale de $O(n^4L)$ opérations. Karmarkar a même été capable de la réduire à $O(n^{3.5}L)$ opérations en utilisant un processus de mise à jour partiel. Ces méthodes projectives n'ont pas été présentées seulement par Karmarkar mais aussi par Todd et Burrell [59], Anstreicher [4][5], Gay [21] et Ye et Kojima [71]. Ces chercheurs ont suggéré des modifications de l'algorithme original de Karmarkar afin d'aboutir à un algorithme plus efficace et plus utile dans la pratique. Ces méthodes ont été mises en oeuvre hors des laboratoires Bell par Adler et al [2][1], Gill et al [22], Monma et Morton [45] et McShane et al. [40], ces différentes mises en oeuvre ont donné de bons résultats prouvant leur supériorité par rapport aux autres méthodes (simplexe, ellipsoïde...). Ce chapitre est structuré comme suit : après avoir exposé le principe, on propose l'algorithme projectif de Karmarkar. Ensuite, on étudie sa complexité suivie de l'approche de Todd et Burrell. Dans une autre section, on présente la mise en oeuvre. Enfin, on montre comment transformer un problème général de programmation linéaire sous la forme donnée par Karmarkar.

3.2 Principe général de l'algorithme de Karmarkar

L'étude sera faite à partir du problème particulier suivant :

$$(P_1) : \begin{cases} \min c^T x \\ Ax = 0 \\ e^T x = n \\ x \geq 0 \end{cases}$$

où $e = (1, \dots, 1)$, A est une matrice $(m \times n)$, c et $x \in \mathbb{R}^n$. On suppose que :

1. La matrice A est de rang maximal.

2. La valeur optimale du problème (P_1) est égale à 0.
3. $Ae = 0$ (ce qui implique que $x^0 = e$ est une solution réalisable pour (P_1)).
4. $c^T e > 0$.

On dénote :

- $\Delta = \{x \in \mathbb{R}^n, e^t x = n, x \geq 0\}$
- $F_{P_1} = \{x \in \mathbb{R}^n, Ax = 0, e^T x = n, x \geq 0\}$ l'ensemble des solutions réalisables du problème (P_1) .
- $\overset{\circ}{F}_{P_1} = \{x \in \mathbb{R}^n, Ax = 0, e^T x = n, x > 0\}$ l'intérieur de F_{P_1} ; $\overset{\circ}{F}_{P_1}$ désigne l'ensemble des solutions strictement réalisables.

Remarque 5 *On a supposé que $c^T e > 0$, car sinon, on s'arrête immédiatement avec e optimal. Ceci implique que $c^T x$, la valeur de l'objectif, n'est pas constante sur la région réalisable; et donc, elle est strictement positive pour chaque point de $\overset{\circ}{F}_{P_1}$.*

3.2.1 Principe de la méthode

L'algorithme de Karmarkar est une procédure itérative qui consiste à générer une suite $\{x^k\}$ de points strictement réalisables et vérifiant :

$$\begin{cases} x^0 = e \\ c^T x^k \leq (\exp \frac{-k}{5n}) c^T x^0 \quad \forall k \geq 1 \end{cases} .$$

Remarques

1. La suite engendrée par l'algorithme vérifie :

$$c^T x^k \leq (\exp \frac{-k}{5n}) c^T x^0 \quad \forall k \geq 1, \tag{3.1}$$

ceci est équivalent à :

$$n \log c^T x^k \leq n \log c^T x^0 - \frac{k}{5} \quad \forall k \geq 1.$$

Afin de mesurer le progrès d'une itération à l'autre, Karmarkar a introduit une fonction auxiliaire appelée fonction de potentiel définie par :

$$F(x) = f(x, c) = n \log c^T x - \sum_{j=1}^n \log x_j = \sum_{j=1}^n \log \frac{c^T x}{x_j}$$

2. Pour passer de l'itération k à l'itération $k + 1$, Karmarkar a utilisé un changement de variable ou, d'une façon plus précise la transformation projective définie par :

$$x \xrightarrow{T_P} \hat{x} = \frac{n(X^k)^{-1}x}{e^T(X^k)^{-1}x}$$

où X^k est une matrice diagonale dont les composantes sont les $x_i, i = 1, \dots, k$.

Propriétés de T_P

La transformation projective présente les propriétés suivantes:

- Elle laisse Δ invariant;
- Elle transforme x^k en e et $F_{P_1}^\circ$ en $\{x \in \Delta, \hat{A}x = 0\}$ où $\hat{A} = AX^k$.
- Elle est aussi bijective et la transformation inverse est donnée par :

$$(T_P)^{-1}(\hat{x}) = \frac{nX^k\hat{x}}{e^T X^k \hat{x}}.$$

De plus, elle garde la même forme pour la fonction de potentiel dans le nouvel espace.

En effet :

Lemme 4 soient x et \hat{x} définis comme précédemment et soit $\hat{c} = X^k c$ alors on a :

$$f(\hat{x}, \hat{c}) = f(x, c) + \log \det X^k.$$

Preuve

$$\text{On a : } f(\hat{x}, \hat{c}) = \sum_{j=1}^n \log \frac{\hat{c}^T \hat{x}}{\hat{x}_j}.$$

Or,

$$\hat{c}^T \hat{x} = \frac{c^T X^k n X^{k-1} x}{e^T X^{k-1} x} = \frac{n c^T x}{e^T (X^k)^{-1} x}$$

et

$$\hat{x}_j = n \frac{((X^k)^{-1} x)_j}{e^T X^{k-1} x} = \frac{n x_j}{e^T (X^k)^{-1} x x_j}$$

donc:

$$\frac{\hat{c}^T \hat{x}}{\hat{x}_j} = \frac{c^T x}{x_j} x_j^k$$

et par suite

$$\log \frac{\hat{c}^T \hat{x}}{\hat{x}_j} = \log \frac{c^T x}{x_j} + \log x_j^k$$

donc

$$\sum_{j=1}^n \log \frac{\hat{c}^T \hat{x}}{\hat{x}_j} = \sum_{j=1}^n \log \frac{c^T x}{x_j} + \sum_{j=1}^n \log x_j^k$$

or

$$\sum_{j=1}^n \log x_j^k = \log \prod_{j=1}^n x_j^k = \log \det X^k$$

d'où le résultat. \square

Donc, pour réduire \hat{f} par une constante à partir de sa valeur en e , il suffit de réduire f par la même constante en utilisant l'étape correspondante dans l'espace original et réciproquement.

Lemme 5 Soit $x \in F_{P_1}$ satisfaisant : $f(x) = f(e) - \gamma$, alors $c^T x \leq (\exp \frac{-\gamma}{n}) c^T e$

Preuve

$$\text{On a : } f(e) = n \log c^T e - \sum_{j=1}^n \log 1 = n \log c^T e.$$

Or, par hypothèse :

$$f(x) = f(e) - \gamma$$

donc :

$$n \log c^T x \leq n \log c^T e - \gamma + \sum_{j=1}^n \log x_j.$$

Les x_j sont des nombres positifs de moyenne arithmétique égale à 1, (car $e^T x = n$) donc leur moyenne géométrique est au plus 1. Ceci implique que $\sum_{j=1}^n \log x_j \leq 0$ et alors : $\log c^T x \leq \log c^T e - \frac{\gamma}{n}$ et par la suite $c^T x \leq (\exp \frac{-\gamma}{n}) c^T e$. \square

L'inégalité (3.1) résulte du lemme 5 si on montre que f diminue à chaque itération par au moins $\frac{1}{5}$. Par le lemme 4, il suffit seulement de diminuer \hat{f} par au moins $\frac{1}{5}$ à partir de sa valeur en e . Pour déterminer une telle diminution, plaçons nous dans le nouvel espace. Considérons la matrice $\hat{B} = \begin{bmatrix} \hat{A} \\ e^T \end{bmatrix}$ et soit $P_{\hat{B}}$: la projection sur le noyau de \hat{B} . Puisque A est de plein rang alors \hat{A} l'est aussi (car la matrice X^k est non singulière) et comme $\hat{A}e = AX^k e = Ax^k = 0$ alors la matrice \hat{B} est de rang maximal. Ainsi $P_{\hat{B}} = P_{\hat{A}} P = P P_{\hat{A}}$ où P est la projection sur $\{x \in \mathbb{R}^n : e^T x = 0\}$. Pour obtenir une diminution convenable de \hat{f} , il suffit de choisir comme direction :

$$\hat{d}' = -P_{\hat{B}} \nabla \hat{f}(e)$$

or, $\nabla \hat{f}(e) = \frac{n\hat{c}}{\hat{c}^T e} - e$. Puisque $\hat{A}e = 0$, donc $P_{\hat{A}}(e) = 0$ et par la suite $P_{\hat{B}}(e) = 0$ d'où $\hat{d}' = P_{\hat{B}}(\frac{n\hat{c}}{\hat{c}^T e}) = -\frac{n}{\hat{c}^T e} P_{\hat{B}}(\hat{c})$. En posant $\check{d} = -P_{\hat{B}}(\hat{c})$ on aura $\hat{d}' = -\frac{n}{\hat{c}^T e} \check{d}$. Karmarkar a montré qu'une réduction constante de f est obtenue en se déplaçant dans le nouvel espace du point e au point $\hat{x} = e + \alpha \frac{\check{d}}{\|\check{d}\|}$.

3.2.2 Détermination du point x^{k+1}

Soit x^k le point obtenu à l'itération k , alors, x^{k+1} est donné par :

$$x^{k+1} = \frac{n\tilde{x}^{k+1}}{e^T \tilde{x}^{k+1}}$$

où $\tilde{x}^{k+1} = x^k + \alpha_k d^k$, α_k et d^k sont définis ci-dessous. Puisque $T_P(x^k) = e$, alors on est au point e dans le nouvel espace à l'itération k . On se déplace dans ce dernier espace suivant la direction $\hat{d}^k = \frac{d}{\|d\|}$ et on obtient le nouveau point : $\hat{x}^{k+1} = e + \alpha_k \hat{d}^k$ avec α_k une longueur de pas convenable assurant la diminution voulue de \hat{f} . Or $x^{k+1} = (T_P)^{-1}(\hat{x}^{k+1}) = \frac{n x^k \hat{x}^{k+1}}{e^T X^k \hat{x}^{k+1}} = \frac{n X^k e + n \alpha_k X^k \hat{d}^k}{e^T X^k e + e^T \alpha_k X^k \hat{d}^k} = \frac{n(x^k + \alpha_k X^k \hat{d}^k)}{e^T(x^k + \alpha_k X^k \hat{d}^k)}$

En posant : $d^k = X^k \hat{d}^k$ et $\tilde{x}^{k+1} = x^k + \alpha_k d^k$, on aura $x^{k+1} = \frac{n \tilde{x}^{k+1}}{e^T \tilde{x}^{k+1}}$. On montrera plus loin que $\alpha_k = \frac{1}{3}$ garantit une diminution de $\frac{1}{5}$ de \hat{f} .

3.2.3 L'algorithme de Karmarkar

Dans ce qui précède, on a montré les mécanismes d'une itération de l'algorithme de Karmarkar. Comme nous le montrons plus loin, la suite de points $\{x^k\}$ engendrée par la méthode est telle que :

$$\lim_{k \rightarrow +\infty} c^T x^k = 0.$$

En pratique, on arrête d'itérer quand la valeur de l'objectif devient suffisamment proche de zéro. Afin de préciser le sens du "suffisamment proche de zéro", on va définir la longueur du problème. Karmarkar définit la constante L_1 , comme borne inférieure de la longueur des données de son problème (P_1) , par :

$$L_1 = 1 + \log |D_{\max}| + \log(1 + |C_{j_{\max}}|)$$

avec $D_{\max} = \max\{|\det B|; B \text{ base de } A\}$ et $C_{j_{\max}} = \max_j |C_j|$.

Si $\alpha = \frac{n-1}{3n}$, alors l'algorithme fournit une solution réalisable vérifiant \bar{x} telle que $c^T \bar{x} < 2^{-L_1}$ en au moins $10nL_1$ itérations. Une fois \bar{x} trouvé, on peut déterminer le point extrême optimal en utilisant la procédure de purification présentée dans le chapitre 1.

Algorithme de Karmarkar

– Initialisation :

– Calculer $L_1 = 1 + \log |D_{\max}| + \log(1 + |C_{j_{\max}}|)$;

– Poser $k = 0$ et prendre $x^k = (1, \dots, 1)^T$;

– α suffisamment petit.

– Étape principale :

– Si $c^T x^k < 2^{-L_1}$, utiliser la procédure décrite précédemment pour trouver un point extrême optimal à partir de x^k et c'est terminé;

– Sinon :

1. Définir :

$$X^k = \text{diag}(x_1^k, \dots, x_n^k), \hat{A} = AX^k \text{ et } \hat{B}^k = \begin{bmatrix} \hat{A} \\ e^T \end{bmatrix}$$

2. Calculer

$$\hat{d}^k = -P_{\hat{B}^k}(X^k c); d^k = X^k \hat{d}^k \text{ et } \tilde{x}^{k+1} = x^k + \alpha \frac{d^k}{\|\hat{d}^k\|}.$$

$$\text{Poser } x^{k+1} = \frac{n\tilde{x}^{k+1}}{e^T \tilde{x}^{k+1}}$$

3. Faire $k = k + 1$ et aller à l'étape principale.

Remarque 6 1. Le choix de α est important parce qu'il gouverne en quelque sorte la valeur de γ (la diminution de la fonction de potentiel);

2. Dans son premier article [32], Karmarkar a choisi $\alpha = \frac{1}{4\sqrt{n(n-1)}}$ et il a abouti à une réduction $\gamma \geq \frac{1}{8}$, alors que Todd et Burrell ont pris $\alpha = \frac{1}{3}$ et ont obtenu exactement $\gamma = \frac{1}{5}$;

3. Todd conjecture qu'en utilisant, α_k le point stationnaire de la fonction $\Phi(\alpha) = f(x^k + \alpha d^k)$, on peut améliorer la convergence. Ceci est équivalent à prendre un pas optimal dans la direction d^k par rapport à la fonction de potentiel f .

En se basant sur l'article de Todd et Burrell [59], on analyse en détails la dérivation pour $\alpha = \frac{1}{3}$. Pour faciliter la présentation, on laisse tomber l'indice k et on considère le problème général suivant :

$$(\bar{P}) \begin{cases} \min \bar{c}^T x \\ \bar{A}x = 0 \\ e^T x = n \\ x \geq 0 \end{cases}$$

satisfaisant les hypothèses 1.- 4. de 3.2.

Remarque 7 Pour retrouver le problème (P_1) on remplace \bar{c} par c et \bar{A} par A et pour retrouver le sous-problème associé à l'itération k on remplace \bar{c} par \hat{c}_k et \bar{A} par \hat{A}_k .

Le dual de (\bar{P}) s'écrit :

$$(\bar{D}) \begin{cases} \max ns \\ \bar{A}^T y + es \leq \bar{c}. \end{cases}$$

Notons que $\forall y \in \mathbb{R}^m$, (y, s) avec $s = \min_{j \in \{1, 2, \dots, n\}} (\bar{c} - \bar{A}^T y)_j$ est une solution réalisable de (\bar{D}) . Or, la valeur optimale de \bar{P} est nulle (d'après la deuxième hypothèse) donc, $ns \leq 0 \forall (y, s)$ réalisables pour \bar{D} et par la suite $s \leq 0 \forall (y, s)$ réalisables.

L'idée est de montrer que le pas de Karmarkar assure une réduction convenable de $\bar{f}(x) = n \log(\bar{c}^T x) - \sum_j \log x_j$ et de vérifier que le terme $-\sum_j \log x_j$ n'augmente pas beaucoup f .

Lemme 6 Soit $\bar{d} = -P_{\bar{B}}(\bar{c})$ avec $\bar{B} = \begin{bmatrix} \bar{A} \\ e^T \end{bmatrix}$, $y = (\bar{A}\bar{A}^T)^{-1}\bar{A}\bar{c}$ et $s = \min_{j \in \{1, 2, \dots, n\}} (\bar{c} - \bar{A}^T y)_j$ alors :

$$\bar{c}^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|} \right) \leq \left(1 - \frac{\alpha}{n} \right) \bar{c}^T e + \frac{\alpha}{n} ns.$$

Preuve

On a $\|\bar{d}\|^2 = \bar{d}^T \bar{d} = \bar{c}^T P_{\bar{B}}(\bar{c}) = -\bar{c}^T \bar{d}$ donc $\bar{c}^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|} \right) = \bar{c}^T e - \alpha \|\bar{d}\|$. Il suffit alors de montrer que : $\|\bar{d}\| \geq \frac{\bar{c}^T e}{n} - s$.

On a $\bar{d} = -P_{\bar{B}}(\bar{c}) = -PP_{\bar{A}}(\bar{c}) = -P(\bar{c} - \bar{A}^T y) = -(\bar{c} - \bar{A}^T y - ee^T(\frac{\bar{c} - \bar{A}^T y}{n}))$ et puisque $Ae = 0$, on aura alors $\bar{d} = -(\bar{c} - \bar{A}^T y) + (\frac{\bar{c}^T e}{n})e$. Or, $\bar{c}^T e \geq V(P_1) \geq ns$, donc $\frac{\bar{c}^T e}{n} > s$. Pour certain i , on a : $s = (\bar{c} - \bar{A}^T y)_i$ donc, $\bar{d}_i = \frac{\bar{c}^T e}{n} - s \geq 0$ et comme $\|\bar{d}\| \geq |\bar{d}_i| = \frac{\bar{c}^T e}{n} - s$. D'où le résultat. \square

Corollaire 1 Avec les mêmes notations que le lemme précédent, si $0 < \alpha < 1$, alors :

$$n \log \bar{c}^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|} \right) \leq n \log(\bar{c}^T e) - \alpha.$$

Preuve

Puisque $ns \leq 0$ donc $\bar{c}^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|} \right) \leq \left(1 - \frac{\alpha}{n} \right) \bar{c}^T e$ (d'après le lemme 6). Or, $\log\left(1 - \frac{\alpha}{n}\right) \leq -\frac{\alpha}{n}$ pour tout $\alpha \in [0, 1]$ et la fonction \log est croissante donc on obtient : $\log \bar{c}^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|} \right) \leq \log(\bar{c}^T e) - \frac{\alpha}{n}$. En multipliant cette inégalité par n , on obtient le résultat. \square

Lemme 7 Si $|\epsilon| \leq \alpha < 1$, alors : $\epsilon - \frac{\epsilon^2}{2(1-\alpha)^2} \leq \log(1+\epsilon) \leq \epsilon$

Preuve

Voir [32] \square .

Lemme 8 Si $\|x - e\| \leq \alpha < 1$, $e^T x = n$, alors : $0 \leq -\sum_j \log x_j \leq \frac{\alpha^2}{2(1-\alpha)^2}$

Preuve

Voir [32] \square .

Théorème 3 Avec les même notations que le lemme 3, on a :

$$f\left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|}\right) \leq f(e) - \alpha + \frac{\alpha^2}{2(1-\alpha)^2}.$$

En particulier, si $\alpha = \frac{1}{3}$, alors : $f\left(e + \frac{1}{3} \frac{\bar{d}}{\|\bar{d}\|}\right) \leq f(e) - \frac{1}{5}$.

Preuve

On a :

$$\begin{aligned} f\left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|}\right) &= n \log c^T \left(e + \alpha \frac{\bar{d}}{\|\bar{d}\|}\right) - \sum_{j=1}^n \log\left(1 + \alpha \frac{\bar{d}_j}{\|\bar{d}\|}\right) \\ &\leq n \log c^T e - \alpha - \sum_{j=1}^n \log\left(1 + \alpha \frac{\bar{d}_j}{\|\bar{d}\|}\right) \quad (\text{d'après le corollaire 1}) \\ &\leq n \log c^T e - \alpha + \frac{\alpha^2}{2(1-\alpha)^2} \quad (\text{d'après le lemme 8}) \\ &= f(e) - \alpha + \frac{\alpha^2}{2(1-\alpha)^2} \end{aligned}$$

En particulier, si $\alpha = \frac{1}{3}$, on obtient : $f\left(e + \frac{1}{3} \frac{\bar{d}}{\|\bar{d}\|}\right) \leq f(e) - \frac{1}{5}$. \square

Théorème 4 Soit $\{x^k\}$ la suite engendrée par l'algorithme de Karmarkar, alors :

$$f(x^k) \leq f(x^0) - \frac{k}{5}$$

Preuve

D'après le théorème précédent: $f(\hat{x}^{k+1}) - f(e) < -\frac{1}{5}$ (dans le nouvel espace). Il s'ensuit que: $f(x^{k+1}) - f(x^k) < -\frac{1}{5}$ (dans l'espace original) et alors: $f(x^k) \leq f(x^0) - \frac{k}{5}$ \square

Corollaire 2 Soit $\{x^k\}$ la suite engendrée par l'algorithme de Karmarkar, alors:

$$c^T x^k \leq \left(\exp \frac{-k}{5n}\right) c^T x^0 \quad \forall k \geq 1.$$

Preuve

D'après le théorème 4, on a:

$$n \log c^T x^k - \sum_{j=1}^n \log x_j^k \leq n \log c^T x^0 - \sum_{j=1}^n \log x_j^0 - \frac{1}{5}.$$

Or, pour $x \in \Delta$ on a $\sum_{j=1}^n \log x_j \leq \sum_{j=1}^n \log x_j^0 = \sum_{j=1}^n \log 1 = 0$, d'où $n \log c^T x \leq n \log c^T x^0 - \frac{k}{5}$ ou encore $c^T x^k \leq \left(\exp \frac{-k}{5n}\right) c^T x^0$. Par conséquent, $\lim_{k \rightarrow +\infty} c^T x^k = 0$ \square

Remarque 8 L'algorithme de Karmarkar présente deux inconvénients majeurs:

- Il exige la connaissance de la valeur optimale de l'objectif;
- Il n'engendre pas une solution optimale duale.

Pour cela Todd et Burrell ont proposé une méthode polynomiale qui utilise le dual du problème (P_1) [59]. Leur méthode n'engendre pas seulement une suite de points réalisables, mais aussi une suite de bornes inférieures de la valeur de l'objectif qui converge vers la valeur optimale.

Nous détaillons cette approche dans la prochaine section:

3.2.4 L'approche de Todd et Burrell : Dualité

Le lemme 6 nous met sur la piste de cette approche. Considérons alors la suite $\{x^k\}$ engendrée par l'algorithme de Karmarkar et la suite associée $\{(y^k, s^k)\}$ comme suit :

$$y^k = (AX^{k^2}A^T)^{-1}AX^k c, \quad s^k = \min_{j \in \{0, 1, \dots, n\}} (c - A^T y^k)_j.$$

Théorème 5 *Supposons que $\{x^k\}$ converge vers une solution de base non dégénérée x^* de (P_1) , alors $\{(y^k, s^k)\}$ converge vers une solution optimale de (D_1) .*

Preuve

Voir [59]. \square

Dualité revue et élimination de la deuxième hypothèse

On sait que :

- La deuxième hypothèse (la valeur optimale est nulle) joue un rôle primordial dans le corollaire 1 pour pouvoir affirmer la réduction de f dans le théorème 1 ;
- Si on connaît que $V(P_1) = ns^*$ pour s^* connue (s^* désigne la solution optimale de (D_1)) alors la valeur optimale de (\bar{P}) serait égale à zéro avec $\bar{A} = A$ et $\bar{c} = c - s^*e$. En effet : $\forall x \in F_{P_1}$, on a $\bar{c}^T x = c^T x - s^*e^T x = c^T x - ns^*$; et alors, on peut appliquer l'algorithme en remplaçant c par $c - s^*e$. Mais la $V(P_1)$, et par conséquent s^* sont inconnus. Pour cela, nous allons utiliser une borne inférieure s^k de s^* et $c(s^k) = c - s^k e$ au lieu de c à l'itération k de l'algorithme.

La nouvelle version de l'algorithme construit deux suites de points $\{x^k\}$ et $\{(y^k, s^k)\}$ ayant les propriétés suivantes :

$$\mathcal{I} : \begin{cases} x^k \in \overset{\circ}{F}_{P_1}, y^k \in \mathbb{R}^m, s^k = \min_{j \in \{1, 2, \dots, n\}} (\bar{c} - \bar{A}^T y^k)_j \\ f(x^k, c(s^k)) \leq f(x^0, c(s^k)) - \frac{k}{5} \quad \forall k \in \mathbb{N}. \end{cases}$$

Pour $k = 0$, on choisit $x^0 = e$, $y^0 = (AA^T)^{-1}Ac$ et le s^0 correspondant

$$(s^0 = \min_{j \in \{1, 2, \dots, n\}} (\bar{c} - \bar{A}^T y^0)_j).$$

Il est trivial de voir que les propriétés sont satisfaites pour $k = 0$.

Remarque 9 Si $c - A^T y^0$ est un multiple de e , alors $c^T x$ est constant sur le domaine réalisable, et on se trouve avec e optimal. Sinon, $c^T x$ n'est pas constant, et alors $c^T x \geq ns$ pour tout x réalisable et pour n'importe quelle borne inférieure ns de $V(P_1)$.

Lemme 9 On a :

$$s^k \leq s^*.$$

Preuve

Par construction (y^k, s^k) est un point réalisable pour (D_1) , et alors : $s^k \leq s^*$. \square

Lemme 10 Soit $s \leq s' \leq s^*$.

Pour tout $x \in \overset{\circ}{F}_{P_1}$ satisfaisant $c^T x > ns'$ et $f(x^0, c(s)) - f(x, c(s)) \geq \gamma > 0$ on a :

$$f(x^0, c(s')) - f(x, c(s')) \geq \gamma.$$

Preuve

Voir [59] \square .

De ceci, il s'ensuit que : $f(x^k, c(s^*)) \leq f(x^0, c(s^*)) - \frac{k}{5} \quad \forall k \in \mathbb{N}$,

d'où la convergence de l'algorithme comme précédemment.

Maintenant, on montre comment trouver x^{k+1} , y^{k+1} et s^{k+1} satisfaisant les propriétés mentionnées précédemment. Pour cela, posons: $y = (A(X^k)^2 A^T)^{-1} A(X^k)^2 c(s^k)$ et $\tilde{s} = \min_{j \in \{1, 2, \dots, n\}} (c - A^T y)_j$.

Deux cas se présentent :

Premier cas: $\tilde{s} \leq s^k$

Alors :

$$\min_{j \in \{1, 2, \dots, n\}} (c - A^T y)_j \leq s^k$$

\Leftrightarrow

$$\min_{j \in \{1, 2, \dots, n\}} (c - s^k e - A^T y)_j \leq 0$$

donc:

$$\min_{j \in \{1, 2, \dots, n\}} (c(s^k) - A^T y)_j \leq 0$$

\Leftrightarrow

$$\min_{j \in \{1, 2, \dots, n\}} (X^k c(s^k) - X^k A^T y)_j \leq 0.$$

En appliquant le lemme 6 avec $\bar{c} = X^k c(s^k)$ et en remplaçant A par $A X^k$, le \tilde{s} correspondant n'est pas positif, et alors $\bar{c}^T x$ peut être réduit par un facteur de $(1 - \frac{\alpha}{n})$ ou α désigne la longueur du pas, par suite la fonction potentielle $f(., c(s^k))$ peut être réduite par au moins $\frac{1}{5}$ en prenant comme direction: $d^k = -X^k P_{B^k} X^k (c - s^k e)$ avec $B^k = \begin{bmatrix} A^k \\ e^T \end{bmatrix}$. Dans ce cas, on prend: $y^{k+1} = y^k$, $s^{k+1} = s^k$ et alors les propriétés \mathcal{I} sont vérifiées pour $k + 1$.

Deuxième cas: $\tilde{s} > s^k$

Alors :

$$\min_{j \in \{1, 2, \dots, n\}} (c(s^k) - A^T y)_j > 0$$

\Leftrightarrow

$$\min_{j \in \{1,2,\dots,n\}} (X^k c(s^k) - X^k A^T y)_j > 0. \quad (3.2)$$

Notons que : $X^k c(s^k) - X^k A^T y = P_{AX^k} X^k c(s^k) = P_{AX^k} (X^k c - s^k x^k)$. Soit $p = P_{AX^k} X^k c$ et $q = P_{AX^k} x^k$. D'après (3.2) on a : $\min_{j \in \{1,2,\dots,n\}} (p - s^k q)_j > 0$. Maintenant, soit $s = \frac{c^T x}{n} \geq s^k$. Notons que $e^T (p - s q) = (P_{AX^k} e)^T (X^k c - s x^k) = e^T (X^k c - s x^k) = c^T x^k - n s = 0$ donc $\min_{j \in \{1,2,\dots,n\}} (p - s^k q)_j \leq 0$ (puisque la somme des composantes s'annule). Alors, il existe s^{k+1} avec $s^k < s^{k+1} \leq s$ tel que $\min_{j \in \{1,2,\dots,n\}} (p - s^k q)_j = 0$. Définissons $y^{k+1} = (AX^{k^2} A^T)^{-1} AX^{k^2} c(s^{k+1})$, et notons que :

$$\min_{j \in \{1,2,\dots,n\}} (c(s^{k+1}) - A^T y^{k+1})_j = \min_{j \in \{1,2,\dots,n\}} (p - s^{k+1} q)_j = 0. \quad (3.3)$$

Donc :

$$\min_{j \in \{1,2,\dots,n\}} (c - A^T y)_j = s^{k+1}.$$

Alors $s^k < s^{k+1} \leq s^*$, et le lemme 6 donne :

$$f(x^0, c(s^{k+1})) - f(x^k, c(s^{k+1})) \geq \frac{k}{5}.$$

De plus d'après (3.3) :

$$\min_{j \in \{1,2,\dots,n\}} (X^k c(s^{k+1}) - X^k A^T y^{k+1})_j = 0.$$

En appliquant le lemme 6 avec $\bar{c} = X^k c(s^{k+1})$ et en remplaçant A par AX^k , le \bar{s} correspondant est nul, et la fonction de potentiel $f(\cdot, c(s^{k+1}))$ peut être diminuée par au moins $\frac{1}{5}$ en prenant comme direction $d^k = -X^k P_{B^k} X^k (c - s^{k+1} e)$.

Algorithme 2

Itération k: x^k, y^k et s^k donnés.

- Calculer : $A^k = AX^k, p = P_{A^k}(X^k c), q = P_{A^k}(x^k)$;

- Si $\min_{j \in \{1, \dots, n\}} (p - s^k q)_j \leq 0$, alors poser : $y^{k+1} = y^k$, $s^{k+1} = s^k$;
 - Sinon :
1. Déterminer s^{k+1} de sorte que $s^{k+1} > s^k$ et $\min_{j \in \{1, 2, \dots, n\}} (p - s^{k+1} q)_j = 0$;
 2. Soit : $y^{k+1} = (A^k A^{kT})^{-1} A X^k c(s^{k+1})$;
 3. Calculer : $\hat{d}^k = -P_{A^k}(p - s^{k+1})$, $d^k = X^k \hat{d}^k$

$$\tilde{x}^{k+1} = x^k + \frac{1}{3} \frac{d^k}{\|d^k\|}, \quad x^{k+1} = \frac{n\tilde{x}^{k+1}}{e^T \tilde{x}^{k+1}}$$

Théorème 6 *L'algorithme 2 engendre une suite $\{x^k\}$ de points réalisables pour (P_1) et une suite $\{(y^k, s^k)\}$ de points réalisables pour (D_1) . De plus, les suites $\{c^T x^k\}$ et $\{ns^k\}$ convergent tous les deux vers ns^* où ns^* désigne la valeur optimale de (P_1) .*

En effet :

$$c^T x^k - ns^* \leq \exp\left(\frac{-k}{5n}\right)(c^T x^0 - ns^*)$$

et

$$ns^* - ns^k \leq \frac{\exp\left(\frac{-k}{5n}\right)}{1 - \exp\left(\frac{-k}{5n}\right)}(c^T x^0 - ns^*)$$

Preuve

D'après ce qui précède, on a :

$$(c^T x^k - ns^*) + (ns^* - ns^k) \leq \exp\left(\frac{-k}{5n}\right)\{(c^T x^0 - ns^*) + (ns^* - ns^k)\}.$$

Par soustraction de $\exp\left(\frac{-k}{5n}\right)(ns^* - ns^k)$ des deux côtés, on en déduit les résultats voulus et, par suite, la convergence désirée. \square

Remarque 10 *Notons que l'inégalité 2 (dans les propriétés des suites $\{x_k\}$ et $\{(y_k, s_k)\}$) assure la convergence rapide du problème (P_1) et de son dual (D_1) (d'après le théorème précédent). Donc, la direction proposée précédemment aide les solutions $\{x^k\}$ et*

$\{(y^k, s^k)\}$ à converger rapidement. Néanmoins, on en déduit qu'il n'existe aucune manière pour montrer que $c^T x^k$ s'approche d'une façon monotone de la solution optimale ou que ns^k augmente d'une façon stricte à chaque itération.

3.3 Mise en oeuvre et complexité

Ici on va discuter comment l'algorithme 2 peut être mis en oeuvre d'une façon efficace. Premièrement, on doit exécuter la factorisation QR de la matrice $(A^k)^T = X^k A^T$ pour toute itération k de la façon suivante: $(A^k)^T = QR = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}$, où $Q = [Q_1, Q_2]$ est une matrice orthogonale, R_1 est une matrice $(m \times m)$ triangulaire supérieure et Q_1 a m colonnes. Cependant, on n'a pas besoin d'avoir Q explicitement, il suffit de pouvoir calculer Qv et $Q^T v$ efficacement. Cette factorisation peut être calculée en $m^2(n - \frac{m}{3})$ opérations en utilisant la méthode de Householder dans le cas où la matrice A^k est dense. Deuxièmement, on calcule $Q^T X^k c$ et $Q^T x^k$ (donc $Q_i^T X^k c$ et $Q_i^T x^k$ pour $i = 1, 2$), et alors, on obtient $u = Q_2 Q_2^T X^k c$ et $v = Q_2 Q_2^T x^k$ en $O(mn)$ opérations. Si y^k et s^k sont mis à jour, s^{k+1} exige alors $O(n)$ opérations et $y^{k+1} = R_1^{-1} Q_1^T (X^k c - s^{k+1} x^k)$ nécessite $O(m^2)$ opérations additionnelles. Finalement $\hat{d}^k, d^k, \tilde{x}^{k+1}$ et x^{k+1} exigent $O(n)$ itérations.

Remarques

1. La factorisation de $(A^k)^T$ est la tâche qui nécessite le plus de temps dans l'algorithme de Karmarkar;
2. La mise en oeuvre de l'algorithme de Karmarkar et ses extensions ont prouvé que, dans la pratique, la méthode projective est plus efficace que le Simplexe surtout pour les problèmes de grande taille (20 à 50 itérations même pour m et n très grands). En effet, plusieurs tests ont été faits par: Ye et Kojima [71], Monma et

Morton [45], Adler et *al.* [2] et plusieurs autres chercheurs qui ont pu mettre en évidence la comparaison remarquable entre ces deux méthodes;

3. La complexité du premier algorithme de Karmarkar est de $O(n^4L)$ opérations arithmétiques ($O(nL)$ itérations et chaque itération exige $O(n^3)$ opérations), mais Karmarkar a été capable de la réduire à $O(n^{3.5}L)$ opérations en utilisant un processus de mise à jour partiel.

3.4 Transformation d'un problème général

Dans cette section, on va montrer comment obtenir un problème de type (P_1) (forme canonique de Karmarkar) à partir d'un problème donné sous forme standard.

Supposons qu'on a à résoudre :

$$(P') \begin{cases} \min r^T t \\ Mt = b \\ t \geq 0 \end{cases}$$

où M est une matrice $(m' \times n')$ de rang maximal.

On suppose que l'addition de la contrainte $e^T t \leq n-2$ a un effet prévisible c'est-à-dire que si le problème (P') admet une solution réalisable optimale, alors elle vérifie $e^T t < n-2$; et si le problème (P') avec cette contrainte ajoutée admet une solution optimale avec $e^T t = n-2$, alors (P') est non borné.

Pour arriver à un problème de type (P_1) , on suit les étapes suivantes :

1. On ajoute la contrainte $e^T t \leq n-2$, qui s'écrit aussi $e^T t + t_1 = n-2$ avec $t_1 \geq 0$,

on obtient :

$$(P'_1) : \begin{cases} \min[r^T, 0] \begin{bmatrix} t \\ t_1 \end{bmatrix} \\ [M, 0] \begin{bmatrix} t \\ t_1 \end{bmatrix} = b \\ e^T t + t_1 = n - 2 \\ t, t_1 \geq 0 \end{cases}$$

2. On introduit une nouvelle variable t_2 de valeur 1 pour déplacer b dans le côté gauche de la contrainte; Alors, on aura :

$$\begin{cases} \min[r^T, 0, 0] \begin{bmatrix} t \\ t_1 \\ t_2 \end{bmatrix} \\ \begin{bmatrix} M & 0 & -b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ e^T t + t_1 + t_2 = n - 1 \\ t, t_1, t_2 \geq 0 \end{cases}$$

3. On introduit une variable artificielle t_3 avec un coût très grand ν pour assurer que le vecteur e soit réalisable, on obtient alors :

$$\begin{cases} \min[r^T, 0, 0, \nu] \begin{bmatrix} t \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} \\ \begin{bmatrix} M & 0 & -b & b - Me \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} t \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ e^T t + t_1 + t_2 + t_3 = n \\ t, t_1, t_2, t_3 \geq 0 \end{cases}$$

4. Finalement on doit éliminer le 1 du terme de droite de la première contrainte en utilisant un multiple de la dernière contrainte et en soustrayant la contrainte

normalisée, on obtient alors :

$$\left\{ \begin{array}{l} \min[r^T, 0, 0, \nu] \begin{bmatrix} t \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} \\ \begin{bmatrix} M & 0 & -b & b - Me \\ -e^T & -1 & n - 1 & -1 \end{bmatrix} \begin{bmatrix} t \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ e^T t + t_1 + t_2 + t_3 = n \\ t, t_1, t_2, t_3 \geq 0 \end{array} \right.$$

D'où la forme exigée, en posant $A = \begin{bmatrix} M & 0 & -b & b - Me \\ -e^T & -1 & n - 1 & -1 \end{bmatrix}$. On a bien $A \in M(m' + 1 \times n' + 3)$ et de rang maximal.

Remarque 11 *On peut choisir $\nu \gg \|\ r \ \|$, et cela ne cause aucune difficulté dans la résolution (pas comme le simplexe où on peut avoir des problèmes considérables).*

CHAPITRE 4

LES MÉTHODES DU POTENTIEL

4.1 Introduction

La méthode de potentiel constitue une autre variante de l'algorithme de Karmarkar. Les algorithmes se distinguent par le fait qu'ils retiennent les caractéristiques de l'algorithme projectif : en particulier, la borne de complexité polynomiale et l'habileté de prendre des grands pas basés sur la fonction de potentiel sans l'utilisation du changement de variables projectif. Le premier algorithme de ce type a été proposé par Gonzaga [25]; puis, une extension de ceci a été proposée par Freund [19], le résultat est entièrement semblable à la forme standard de l'algorithme projectif. Indépendamment, Ye [70] a dérivé un algorithme similaire qui utilise une fonction de potentiel primale-duale pour obtenir un algorithme de complexité meilleure que ceux mentionnés précédemment par un facteur de \sqrt{n} . Une variante de l'algorithme de Ye a été donnée par Freund [19]. Récemment, Gonzaga [27] a montré que l'amélioration de la complexité proposée par Ye peut être obtenue sans l'addition d'un terme de pénalité duale dans la fonction de potentiel. Malheureusement, tous

ces algorithmes exigent une solution initiale réalisable; donc, une variable artificielle avec un cout grand "M" doit être ajoutée à ce problème. Le premier essai pour développer un algorithme potentiel conditionné qui n'exige pas la réalisabilité a été fait par Freund [20], utilisant une modification (translation) du terme de pénalité dans la fonction de potentiel. Anstreicher [6], à son tour, a développé une extension simple de l'algorithme potentiel conditionné, celui-ci obtient simultanément la réalisabilité et l'optimalité sans l'addition du terme "M". Sa méthode a été considérée comme une extension de l'algorithme de Gonzaga [25] et elle a été améliorée aussi par Todd [62]. Dernièrement, Anstreicher et Bosch [3] ont proposé un algorithme dont la complexité est meilleure que celle des algorithmes mentionnés précédemment. C'est le premier algorithme à $O(n^3L)$ opérations pour des problèmes de programmation linéaire. Ils montrent que l'utilisation de la règle de "Goldstein-Armijo" pour sauvegarder la direction cherchée durant les pas primals est suffisante pour garder une moyenne de complexité de $O(n^{2.5})$ opérations par itération, ils étendent le pas dual pour appliquer la mise à jour partielle, ainsi la complexité totale de leur algorithme est seulement $O(n^3L)$ opérations. Dans ce chapitre, on commence par donner le principe de la méthode; puis, on décrit les principaux types d'algorithmes, à savoir l'algorithme potentiel primal et l'algorithme potentiel primal-dual ainsi que leur complexité.

4.2 Principe général de la méthode

On considère le problème (P) ainsi que son dual (D). Les algorithmes de cette méthode sont simplement des algorithmes de descente de plus forte pente avec conditionnement affine appliqué à une fonction de potentiel.

4.2.1 Les fonctions de potentiel

Karmarkar a été le premier à introduire une fonction de potentiel en programmation linéaire. La fonction de potentiel primale qu'il a définie est :

$$f_{KP}(x, v_L) = q \log(c^T x - v_L) - \sum_{j=1}^n \log x_j$$

où v_L est une borne inférieure de la valeur optimale v^* et $q = n + 1$. Cette fonction a été utilisée par [71] et [25], avec d'autres valeurs de q , tandis que sa fonction duale était :

$$f_{KD}(y, v_U) = q \log(v_U - b^T y) - \sum_{j=1}^n \log s_j,$$

où v_U est une borne supérieure de v^* . Ye et Todd [60], à leur tour, ont proposé la fonction de potentiel primale-duale suivante :

$$f_{YT}(x, s) = q \log(x^T s) - \sum_{j=1}^n \log x_j - \sum_{j=1}^n \log s_j,$$

où $q = 2n$ dans [61] et $q = n + \sqrt{n}$ dans [60]. Cette fonction a été à la base de l'algorithme de Ye [70], elle a été utilisée par Tanabe dans [56]. Ces fonctions ont pour rôle primordial de mesurer le progrès de l'algorithme (analyse et complexité), elles peuvent aussi être utilisées dans la recherche des directions. Les fonctions de potentiel ne sont pas convexes; cependant, elles admettent un minimum unique pour n'importe quelle borne v ($v = v_L$ ou $v = z_U$) et quel que soit $q \geq n$. Ceci est une conséquence de la convexité stricte de la fonction de potentiel multiplicative comme l'ont montré Iri et Imai [30].

Pour définir les fonctions de potentiel primales, on a besoin des bornes inférieures de la valeur optimale; donc, il sera intéressant de donner une procédure pour la recherche de telles bornes.

Les bornes inférieures

Todd considère le problème donné par:

$$(P_\xi) \begin{cases} \min c^T x \\ Ax = b \\ \xi^T x = 0 \\ x \geq 0 \end{cases}$$

avec A est une matrice ($m \times n$) de rang maximal; $0 < m < n$; c , ξ et $x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$, pour lequel il est plus facile de générer une borne inférieure que pour le problème standard. Cependant tout problème standard peut se transformer facilement en (P_ξ) (Voir la remarque ci-dessous). Soit alors le problème (P_ξ) . On définit :

$$- \overset{\circ}{F}_{P_\xi} = \{x \in \mathbb{R}^n, Ax = b, \xi^T x = 0, x > 0\};$$

$$- F'_{P_\xi} = \{x \in \mathbb{R}^n, Ax = b, x > 0\}.$$

On suppose qu'on connaît un point de $\overset{\circ}{F}_{P_\xi}$. Par changement de variable, on peut supposer que c'est le point $e = (1, \dots, 1)^T$. Sans perte de généralité, on suppose que $\xi^T e \geq 0$, sinon on peut changer le signe de ξ . Si $\xi^T e = 0$, alors $e \in \overset{\circ}{F}_{P_\xi}$, et on peut continuer avec l'algorithme de potentiel primal donné dans la prochaine section.

Dans ce qui suit, on suppose que $\xi^T e > 0$.

Remarque 12 *Étant donné le problème suivant :*

$$(P) \begin{cases} \min \tilde{c}^T \tilde{x} \\ \tilde{A}\tilde{x} = \tilde{b} \\ \tilde{x} \geq 0 \end{cases}$$

On définit: $x^T = (\tilde{x}^T, \tilde{x}_+)$, $c^T = (\tilde{c}^T, 0)$, $\xi^T = (0, 1)$ et $A = (\tilde{A}, \tilde{b} - \tilde{A}\tilde{e})$ où \tilde{e} est un vecteur dont les composantes sont tous égales à 1 ayant la même dimension que \tilde{x} . Alors,

le problème (P_ξ) est équivalent au problème (P) , puisque la contrainte $\xi^T x = 0$ oblige la dernière variable à être nulle, $e \in F'_{P_\xi}$ et $\xi^T e = 1 > 0$.

On suppose que l'itération courante est e et la borne inférieure est v_L (v_L peut être égale à $-\infty$). On pose $\Delta_1 = \xi^T e$, par hypothèse, on a $\Delta_1 > 0$.

En écrivant $x = x' + e$, le problème (P_ξ) devient :

$$(P'_\xi) \left\{ \begin{array}{l} \min c^T x' + c^T e \\ Ax' = 0 \\ \xi^T x' = -\Delta_1 \\ x' \geq -e \end{array} \right.$$

Donc $x' \in \mathcal{N}(A)$ où $\mathcal{N}(A)$ désigne le noyau de la matrice A . Pour tout vecteur u , on désigne par u_p sa projection sur $\mathcal{N}(A)$, donc $x'_p = x'$. En utilisant les propriétés de la projection on a $u^T x' = u^T x'_p = u_p^T x'$ pour tout vecteur u . Donc, en remplaçant c et ξ par c_p et ξ_p dans (P'_ξ) et en ajoutant la contrainte $(e - e_p)^T x' = 0$, on aura le problème équivalent suivant :

$$(P''_\xi) \left\{ \begin{array}{l} \min c_p^T x' + c^T e \\ Ax' = 0 \\ \xi_p^T x' = -\Delta_1 \\ (e - e_p)^T x' = 0 \\ x' \geq -e \end{array} \right.$$

En utilisant la relaxation de Fraley [18], on obtient :

$$(FRP) \left\{ \begin{array}{l} \min c_p^T x' + c^T e \\ \xi_p^T x' = -\Delta_1 \\ (e - e_p)^T x' = 0 \\ x' \geq -e \end{array} \right.$$

Son dual est :

$$(FRD) \left\{ \begin{array}{l} \max -\Delta_1 \lambda - e^T s + c^T e \\ \xi_p \lambda + (e - e_p) \tau + s = c_p \\ s \geq 0 \end{array} \right.$$

Ce problème est lié au dual (D_ξ) de (P_ξ) qui est :

$$(D_\xi) \left\{ \begin{array}{l} \max b^T y \\ A^T y + \xi \lambda + s = c \\ s \geq 0. \end{array} \right.$$

En effet, puisque $b = Ae$, la fonction objectif du dual peut s'écrire: $e^T A^T y = c^T e - \Delta_1 \lambda - e^T s$, comme dans (FRD) . Aussi, pour tout vecteur u , $u - u_p \in \mathcal{R}(A^T)$. En écrivant $u - u_p = A^T y_u$, alors le problème (FRD) peut s'écrire :

$$(FRD') \left\{ \begin{array}{l} \max -\Delta_1 \lambda - e^T s + c^T e \\ A^T (y_c - \lambda y_\xi + \tau y_e) + \xi \lambda + s = c \\ s \geq 0. \end{array} \right.$$

Ce problème est clairement une restriction de (D_ξ) , puisque y doit être une fonction spécifiée en λ et τ . Ainsi, la valeur optimale du problème (FRD) est inférieure ou égale à celle du problème (D_ξ) . Simplifions encore (FRD') . Tout d'abord, on élimine la constante $c^T e$ et on met au négatif la fonction objectif pour avoir $\min(\Delta_1 \lambda + e^T s)$. Ceci représente la marge au-dessous de la valeur courante $c^T e$. Notons que cette marge doit être négative puisque e n'est pas réalisable. Après, on élimine s de la fonction objectif en utilisant la première contrainte de (FRD') , on aura alors :

$$(FRD'') \left\{ \begin{array}{l} \min(-\xi_p^T + \Delta_1) \lambda - (e_p^T e - n) + c_p^T e \\ \xi_p \lambda + (e - e_p) \tau \leq c_p. \end{array} \right.$$

Ceci est un problème de programmation linéaire à deux dimensions qui peut être résolu facilement.

- Supposons que (FRD'') a une solution optimale avec une valeur δ , alors (FRD) a une valeur optimale $c^T e - \delta$. Il existe donc une solution réalisable à (D_ξ) avec une valeur de l'objectif égale à $c^T e - \delta$. Alors, $v_L^+ = \max\{v_L, c^T e - \delta\}$ est notre nouvelle borne inférieure;
- Si le problème (FRD'') est non borné, alors (FRD) et (D_ξ) le sont aussi. Dans ce cas, (P_ξ) est non réalisable;
- Si (FRD'') est non réalisable, alors (FRD) l'est aussi et donc on ne peut rien conclure sur (D_ξ) , Dans ce cas, on se contente de $v_L^+ = v_L$.

4.3 Méthode du potentiel primale

Gonzaga est le premier qui a décrit un algorithme potentiel primal [25]. Son algorithme est simplement la méthode de descente de plus forte pente avec conditionnement affine appliqué à la fonction de potentiel primale de Karmarkar :

$$f_{KP}(x, v_L) = q \log(c^T x - v_L) - \sum_{j=1}^n \log x_j$$

il a supposé que la valeur optimale v^* de (P) est nulle. Freund [19] a proposé une extension de l'algorithme de Gonzaga. Cependant, il a utilisé une fonction de potentiel particulière pour mesurer la performance de son algorithme : $f(x) = q \log(x^T s) - \sum_{j=1}^n \log x_j$. Dans cette section, on va proposer un algorithme de potentiel primal et étudier sa polynomialité. Pour faciliter l'analyse, on suppose que la valeur optimale du problème (P) est nulle et on considère la fonction de potentiel suivante :

$$f(x) = q \log(c^T x) - \sum_{j=1}^n \log x_j$$

où $x > 0$ et $q > 0$. Cette fonction est différentiable et on a : $\forall x > 0, \nabla f(x) = \frac{q}{c^T x} c - X^{-1} e$. En particulier, $\nabla f(e) = \frac{q}{c^T e} c - e$.

4.3.1 Algorithme potentiel primal

– Données :

- x^0 strictement réalisable pour (P) ;
- ϵ un paramètre de tolérance strictement positif;

Pour $k = 0, 1, \dots$

- Répéter :

- Conditionnement : $\bar{A} = AX^k$, $\bar{c} = X^k c$, $y = (X^k)^{-1}x$, $X^k = \text{diag}(x_1^k, \dots, x_n^k)$,
 $\bar{f}(y) = q \log(\bar{c}^T y) - \sum_{j=1}^n \log y_j$

- Direction : $h = -P_{\bar{A}} \nabla \bar{f}(y)$ où $P_{\bar{A}} = I - \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A}^T$

- Pas : chercher approximativement $\bar{\lambda} = \arg \min \{ \bar{f}(y + \lambda h); \lambda > 0; y + \lambda h > 0 \}$

- $y^* = y + \bar{\lambda} h$

- $x^{k+1} = X^k y^*$

- $k = k + 1$

- Jusqu'à : $c^T x^k \leq \epsilon$

Remarque 13 Pour $x = X^k y$, $f(x) = \bar{f}(y) - \sum_{j=1}^n \log X_j^k$; donc le conditionnement affine n'a pas d'effet sur la variation de la fonction de potentiel.

Notre but est de prouver que cette fonction de potentiel diminue à chaque itération par un facteur constant indépendant de l'itération. Il suffit alors de montrer qu'il existe une constante α indépendante de k telle que : $\bar{f}(y^*) \leq \bar{f}(e) - \alpha$. Pour cela, on considère le problème :

$$(\bar{P}) \begin{cases} \min \bar{c}^T x \\ \bar{A}x = b \\ x \geq 0 \end{cases}$$

Lemme 11 On considère le problème (\bar{P}) et soit $h = -P_{\bar{A}} \nabla \bar{f}(e)$, alors :

- Si $q \geq n + \sqrt{n}$, alors $\|h\| \geq 1$

- Si $q = n$, alors $\|h\| \geq \frac{1}{2\sqrt{n}}$

Preuve

Voir [25]. \square

Lemme 12 *On considère le problème (\bar{P}) et soit $h = -P_A \nabla \bar{f}(e)$, alors :*

- Si $q \geq n + \sqrt{n}$, alors $\bar{f}(y^*) - \bar{f}(e) \leq -0.1$
- Si $q = n$, alors $\bar{f}(y^*) - \bar{f}(e) \leq \frac{-0.03}{n}$

Preuve

Soit $y^* = e + \lambda^* h$.

Par définition de λ^* , $\forall \lambda > 0$ vérifiant $e + \lambda h > 0$ on a : $\bar{f}(y^*) \leq \bar{f}(e + \lambda h)$.

En particulier, $\forall \lambda \in [0, \frac{0.5}{\|h\|}]$, on a $\lambda \|h\|_\infty \leq 0.5$, donc en utilisant le développement de Taylor, on aura :

$$\bar{f}(y^*) \leq \bar{f}(e + \lambda h) \leq \bar{f}(e) + \lambda \nabla \bar{f}(e)^T h + 2\lambda^2 \|h\|^2.$$

Puisque $h \in \mathcal{N}(\mathcal{A})$, $\nabla \bar{f}(e)^T h = -\|h\|^2$, par la suite : $\bar{f}(y^*) - \bar{f}(e) \leq -\lambda \|h\|^2 (1 - 2\lambda)$.

- Si $q \geq n + \sqrt{n}$, alors par le lemme 1, on a : $\|h\| \geq 1$. Choisissons $\lambda = \frac{0.3}{\|h\|}$ et alors : $\bar{f}(y^*) - \bar{f}(e) \leq -0.3 \|h\| (1 - \frac{0.6}{\|h\|}) \leq -0.1$;
- Si $q = n$, on distingue deux cas :

- Si $\|h\| \geq 1$, alors on choisit $\lambda = \frac{0.3}{\|h\|}$ et on obtient :

$$\begin{aligned} \bar{f}(y^*) - \bar{f}(e) &\leq -0.1 \\ &\leq \frac{0.03}{n} \end{aligned}$$

- Sinon, on prend $\lambda = 0.3$ et alors :

$$\begin{aligned} \bar{f}(y^*) - \bar{f}(e) &\leq -0.12 \|h\|^2 \\ &\leq \frac{-0.12}{4n} \quad (\text{d'après le lemme 1}) \\ &= \frac{0.03}{n}. \end{aligned}$$

Ainsi, on obtient les résultats. \square

Donc, si $q = O(n)$, on peut trouver α vérifiant : $\bar{f}(y^*) - \bar{f}(e) \leq \alpha$ ou encore :

$$f(x^k) - f(x^{k-1}) \leq \alpha \quad \text{pour tout } k \quad (4.1)$$

Lemme 13 *Puisque f satisfait la condition (4.1) alors le problème (P) est résolu en au plus $O(\frac{q}{\alpha}L)$ itérations où L est la longueur des données du problème.*

Preuve

On suppose le problème résolu lorsqu'on trouve un point x^k vérifiant $c^T x^k < 2^{-L}$ ($\epsilon = 2^{-L}$) (à partir de ce point, la solution optimale peut être obtenue par une procédure de purification dans un $O(n^3)$ opérations arithmétiques).

L'ensemble réalisable est borné, donc, il existe une constante positive M telle que pour tout x réalisable, on a : $\sum_{j=1}^n \log x_j \leq M$ ($M \leq O(L)$). Étant donné le point x^0 , on a

$f(x^k) \leq f(x^0) - k\alpha$. Donc : $q \log c^T x^k - \sum_{j=1}^n \log x_j^k \leq f(x^0) - k\alpha$. Par la suite : $q \log c^T x^k \leq$

$M + f(x^0) - k\alpha$, ou encore : $\log c^T x^k \leq \frac{M + f(x^0)}{q} - k\frac{\alpha}{q}$. La condition $\log c^T x^k \leq -L$

est satisfaite pour tout k tel que : $\frac{M + f(x^0)}{q} - k\frac{\alpha}{q} \leq -L$

ou encore $k \geq \frac{\alpha}{q}(L + \frac{M + f(x^0)}{q})$. Puisque $\frac{M + f(x^0)}{q} \leq O(L)$, donc $k = O(\frac{\alpha}{q}L)$. \square

4.3.2 Complexité de l'algorithme

Théorème 7 *On suppose que $q = O(n)$, alors l'algorithme de potentiel primal appliqué au problème (P) possède la propriété suivante :*

- Si $q \geq n + \sqrt{n}$, alors l'algorithme se termine en au plus $O(nL)$ itérations;
- Si $q = n$, alors l'algorithme se termine dans au plus $O(n^2L)$ itérations.

Preuve

On suppose que $q = O(n)$

- Si $q \geq n + \sqrt{n}$, on a : $f(x^k) - f(x^{k-1}) \leq -0.1 \forall k > 0$, (d'après le lemme 13).
donc, l'algorithme se termine dans $O(\frac{q}{0.1}L)$. Puisque $q = O(n)$, on obtient alors le résultat voulu.
- Même raisonnement pour $q = n$. \square

4.4 Méthode de potentiel duale

Cette méthode est similaire à la précédente, sauf qu'ici on utilise la fonction potentielle duale de Karmakar:

$$f_{KD}(y, v_U) = q \log(v_U - b^T y) - \sum_{j=1}^n \log s_j$$

où v_U est une borne supérieure de v^* . De plus, le conditionnement affine est utilisé avec la matrice S^{-1} où $S^{-1} = \text{diag}(s_1^{-1}, \dots, s_n^{-1})$ (s désigne la variable d'écart pour le problème dual).

4.5 Méthode du potentiel primale-duale

La fonction de potentiel primale-duale de Tanabe [56] et de Todd et Ye [60][61] est définie par :

$$\Phi_q(x, s) = q \log(x^T s) - \sum_{j=1}^n \log x_j - \sum_{j=1}^n \log s_j$$

pour $q > n$. Cette fonction est déterminante pour le développement des algorithmes de la méthode du potentiel depuis 1988. Sa relation avec la fonction potentielle primale est

donnée par :

$$\Phi_q(x, s) = f(x, v_L) - \sum_{j=1}^n \log s_j$$

tandis que sa relation avec la fonction duale est donnée par :

$$\Phi_q(x, s) = f(y, v_U) - \sum_{j=1}^n \log x_j.$$

Certains algorithmes basés sur Φ_q traitent les variables primales et duales de façons différentes en utilisant la fonction potentielle seulement pour mesurer le progrès et la convergence de la méthode. D'autres, comme les algorithmes décrits par Todd [62] et Freund [19] déterminent les directions ainsi que les pas pour les variables primales et parfois mettent à jour les variables duales. D'autres aussi, comme les algorithmes de Todd [62] et Gonzaga [25], prennent des itérations dans les deux espaces primals et duals en utilisant une certaine mesure de centralisation pour choisir entre les deux possibilités. Pour faire l'analyse de la méthode, nous allons procéder de la même façon que Kojima et *al.* [37]. On va utiliser la fonction de potentiel Φ_q :

$$\Phi_q(x, s) = q \log(x^T s) - \sum_{j=1}^n \log x_j s_j \quad (4.2)$$

$$= (q - n) \log(x^T s) - \sum_{j=1}^n \log \frac{x_j s_j}{\frac{x^T s}{n}} + n \log n, \quad (4.3)$$

et soit $F_P * F_D = \{(x, s); x \text{ réalisable pour (P) et } s \text{ réalisable pour (D)}\}$ et $\overset{\circ}{F}_P * \overset{\circ}{F}_D$ son intérieur. L'algorithme proposé engendre une suite de points (x^k, y^k, s^k) strictement réalisables et tels que : $\Phi_q(x^k, s^k) \searrow -\infty$.

4.5.1 Algorithme de potentiel primal-dual

– Données :

– Un point $(x^0, y^0, s^0) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$

– $q > n$

Pour $k = 0, 1, \dots$

– Répéter :

– Soit $\sigma_k = \frac{n}{q}$

– Résoudre le système :
$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} dx^k \\ dy^k \\ ds^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{pmatrix}$$
 pour obtenir la direction $(dx^k, dy^k, ds^k)^T$.

– Calculer $\lambda_{max} = \sup\{\lambda \in [0, 1] / (x, s) + \lambda(dx, ds) \geq 0\}$.

– Choisir $\lambda_k = \arg \min_{\lambda \in (0, \lambda_{max})} \Phi_q(x^k + \lambda dx^k, s^k + \lambda ds^k)$.

– $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \lambda_k(dx^k, dy^k, ds^k)$.

– Fin.

4.5.2 Complexité de l'algorithme potentiel primal-dual

Pour déterminer la complexité de l'algorithme on se sert de quelques résultats techniques :

Résultat 1 :

– $\log(1 + \beta) \leq \beta \quad \forall \quad \beta \geq 1$; (avec égalité si et seulement si $XSe = \mu e$)

– $\forall t \in \mathbb{R}^n$ avec $\|t\|_\infty < \tau < 1$ on a :

$$-\sum_{j=1}^n \log(1 + t_j) \leq -e^T t + \frac{\|t\|^2}{2(1 - \tau)}.$$

Preuve

Voir [70]. \square

Résultat 2: Pour tout $(x, s) > 0$, on a : $\Phi_n(x, s) \geq n \log n$ (avec égalité si $XSe = \mu e$) \square

Preuve

$$\text{On a : } \Phi_n(x, s) - n \log n = - \sum_{j=1}^n \log \frac{x_j s_j}{x^T s} \geq - \sum_{j=1}^n \frac{x_j s_j}{\mu} - 1 = -(n - n) = 0 \quad \square$$

Résultat 3:

- La fonction Φ_q est non bornée inférieurement sur son domaine
- Pour tout $(x, y, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_P$, on a : $\mu \leq \exp\left(\frac{\Phi_q(x, s)}{q - n}\right)$ où $\mu = \frac{x^T s}{n}$

Preuve

- On sait que pour tout μ strictement positif, il existe un point réalisable (x_μ, y_μ, s_μ) tel que $(x_\mu)_j (s_\mu)_j = \mu$ pour tout j ; or,

$$\Phi_q(x_\mu, s_\mu) = (q - n) \log(x_\mu^T s_\mu) + \Phi_n(x_\mu s_\mu) \quad (4.4)$$

$$= (q - n) \log n \mu + n \log n \quad (4.5)$$

donc, quand $\mu \rightarrow 0$, $\Phi_q(x_\mu s_\mu) \rightarrow -\infty$.

- On a :

$$\Phi_q(x_\mu, s_\mu) = (q - n) \log x^T s + \Phi_n(x, s) \quad (4.6)$$

$$\geq (q - n) \log \mu + (q - n) \log n + n \log n \quad (4.7)$$

$$\geq (q - n) \log \mu. \quad (4.8)$$

D'où :

$$\mu \leq \exp\left(\frac{\Phi_q(x, s)}{q - n}\right)$$

D'où le résultat. \square

En conséquence : si l'algorithme engendre une suite de points strictement réalisables

(x^k, y^k, s^k) vérifiant $\Phi_q(x^k, s^k) \searrow -\infty$, alors $\mu_k \rightarrow 0$.

Si on arrive à trouver une constante disons $\delta > 0$ telle que pour tout k on a :

$$\Phi_q(x^{k+1}, s^{k+1}) \leq \Phi_q(x^k, s^k) - \delta \quad (4.9)$$

on peut alors déterminer la complexité de l'algorithme 2.

Théorème 8 *Étant donné un point (x^0, y^0, s^0) strictement réalisable. On suppose que l'algorithme 2 engendre une suite de points strictement réalisables satisfaisant (4.9) alors pour tout $\epsilon \in (0, 1)$, il existe un indice K défini par :*

$$K = \lfloor \frac{\Phi_q(x^0, s^0)}{\delta} + \frac{q-n}{\delta} |\log \epsilon| \rfloor \quad (4.10)$$

tel que $\mu_k \leq \epsilon$ pour tout k supérieur ou égale à K .

Remarque 14 $\lfloor x \rfloor$ désigne la partie entière de x .

Preuve

On a :

$$\Phi_q(x^k, s^k) \leq (q-n) \log \mu \quad (4.11)$$

$$= -(q-n) |\log \epsilon| \quad (4.12)$$

D'après la relation 4.9 on a : $\Phi_q(x^k, s^k) \leq \Phi_q(x^0, s^0) - k\delta$ pour tout k . Donc, il suffit que : $\Phi_q(x^0, s^0) - k\delta \leq -(q-n) |\log \epsilon|$ ou encore : $k \geq \frac{\Phi_q(x^0, s^0)}{\delta} + \frac{q-n}{\delta} |\log \epsilon|$ il suffit de prendre $K = \lfloor \frac{\Phi_q(x^0, s^0)}{\delta} + \frac{q-n}{\delta} |\log \epsilon| \rfloor$.

Il reste à déterminer la réduction de la fonction de potentiel. Pour cela, examinons le

déroulement d'une itération. Soit $\lambda \in (0, \lambda_{\max})$, on a :

$$\begin{aligned}
\Phi_q(x + \lambda d_x, s + \lambda d_s) - \Phi_q(x, s) &= q \log \frac{(x + \lambda d_x)^T (s + \lambda d_s)}{x^T s} - \sum_{j=1}^n \log \frac{x_j + \lambda d_{x_j}}{x_j} \\
&- \sum_{j=1}^n \log \frac{s_j + \lambda d_{s_j}}{s_j} \\
&= q \log \left(1 + \lambda \frac{s^T d_x + x^T d_s + \lambda d_x^T d_s}{x^T s} \right) - \sum_{j=1}^n \log \frac{x_j + \lambda d_{x_j}}{x_j} \\
&- \sum_{j=1}^n \log \frac{s_j + \lambda d_{s_j}}{s_j} \\
&= q \log \left(1 + \lambda \frac{s^T d_x + x^T d_s}{x^T s} \right) - \sum_{j=1}^n \log \left(1 + \lambda \frac{d_{x_j}}{x_j} \right) \\
&- \sum_{j=1}^n \log \left(1 + \lambda \frac{d_{s_j}}{s_j} \right) \quad (\text{car } d_x^T d_s = 0)
\end{aligned}$$

Pour assurer que le nouveau point soit dans $\overset{\circ}{F}_P * \overset{\circ}{F}_D$, il faut que (d_x, d_s) et λ satisfassent :

$$\lambda \max(\| X^{-1} d_x \|_{\infty}, \| S^{-1} d_s \|_{\infty}) < \tau$$

où $\tau \in (0, 1)$ une constante à déterminer.

En utilisant le résultat 1, on aura :

$$\begin{aligned}
\Phi_q(x + \lambda d_x, s + \lambda d_s) - \Phi_q(x, s) &\leq q \lambda \frac{s^T d_x + x^T d_s}{x^T s} - \lambda e^T (X^{-1} d_x + S^{-1} d_s) \\
&+ \frac{\lambda^2}{2(1-\tau)} (\| X^{-1} d_x \|^2 + \| S^{-1} d_s \|^2)
\end{aligned}$$

$$\text{Soit } g_1 = q \frac{s^T d_x + x^T d_s}{x^T s} \quad \text{et} \quad g_2 = \frac{1}{(1-\tau)} (\| X^{-1} d_x \|^2 + \| S^{-1} d_s \|^2)$$

On aura :

$$\Phi_q(x + \lambda d_x, s + \lambda d_s) \leq \Phi_q(x, s) + \lambda g_1 + \frac{\lambda^2}{2} g_2$$

Si on pose $g(\lambda) = \Phi_q(x, s) + \lambda g_1 + \frac{\lambda^2}{2} g_2$, on aura :

$$\Phi_q(x + \lambda d_x, s + \lambda d_s) \leq g(\lambda) \quad \forall \lambda \in (0, \lambda_\tau),$$

où $\lambda_\tau \max(\|X^{-1}d_x\|_\infty, \|S^{-1}d_s\|_\infty) = \tau$

Étude de $g(\lambda)$:

On a : $g(\lambda) = \Phi_q(x, s) + \lambda g_1 + \frac{\lambda^2}{2} g_2$

- Comme $g_2 > 0$, alors g est une fonction quadratique convexe;
- pour montrer que g est décroissante en λ , il faut prouver que g_1 est négatif.

Pour cela, on définit les notations suivantes : Étant donné $(x, y, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$, soit :
 $V = (XS)^{\frac{1}{2}}$, $v = Ve$, $D = X^{\frac{1}{2}}S^{\frac{1}{2}}$, $v_{\min} = \min_{1 \leq i \leq n} v_i$, $r = -v + \frac{n}{q}\mu V^{-1}e$. Les relations entre D , v et x sont données par :

$$\|v\| = x^T s = n\mu, \quad X = VD, \quad S = VD^{-1}.$$

Puisque la direction qu'on cherche vérifie : $Sd_x + Xd_s = -XSe + \frac{n}{q}$, on aura alors le système suivant :

$$\begin{cases} Sd_x + Xd_s & = & Vr \\ D^{-1}d_x + Dd_s & = & r \\ X^{-1}d_x + S^{-1}d_s & = & V^{-1}r. \end{cases}$$

Donc, on obtient :

$$\|r\|^2 = \|D^{-1}d_x + Dd_s\|^2 = \|D^{-1}d_x\|^2 + \|Dd_s\|^2 \quad (\text{card}_x^T d_s = 0)$$

par la suite : $\|D^{-1}d_x\| \leq \|r\|$ et $\|Dd_s\| \leq \|r\|$. De plus :

$$\begin{aligned} \|X^{-1}d_x\|^2 + \|S^{-1}d_s\|^2 &\leq \|V^{-1}D^{-1}d_x\|^2 + \|V^{-1}Dd_s\|^2 \\ &= \|V^{-1}\|^2 (\|D^{-1}d_x\|^2 + \|Dd_s\|^2) \\ &\leq \frac{\|r\|^2}{v_{\min}^2} \end{aligned} \tag{4.13}$$

En conséquence :

$$g_2 \leq \frac{\|r\|^2}{(1-\tau)v_{\min}^2}.$$

D'autre part :

$$\begin{aligned} g_1 &= \frac{q}{x^T s} e^T (Sd_x + Xd_s) - e^T (X^{-1}d_x + S^{-1}d_s) \\ &= \frac{q}{n\mu} e^T V r - e^T V^{-1} r \\ &= -\frac{q}{n\mu} \left(-v + \frac{n\mu}{q} V^{-1} e\right)^T r \\ &= -\frac{q}{n\mu} \|r\|^2. \end{aligned}$$

D'où : $\forall r \neq 0$ on a : $g_1 < 0$ et $g_2 > 0$. Or, $\forall \lambda \in (0, \lambda_\tau)$ $g'(\lambda) = g_1 + \lambda g_2 \implies g'(\lambda^*) = 0 \iff \lambda^* = -\frac{g_1}{g_2}$. En conclusion :

$$g(\lambda^*) \leq g(0) \square$$

Lemme 14 $\forall (x, y, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$ et $q > n + \sqrt{n}$, on a :

$$\|r\|^2 \geq \frac{\sqrt{3}}{2v_{\min}} \frac{n\mu}{q}.$$

Preuve

On a

$$\begin{aligned} \frac{q^2}{n^2 \mu^2} \|r\|^2 &= \left\| \frac{q}{n\mu} v - V^{-1} e \right\|^2 \\ &= \frac{q^2}{n^2 \mu^2} v^T v + (V^{-1} e)^T V^{-1} e - 2 \frac{q}{n\mu} v^T V^{-1} e \\ &= (V^{-1} e)^T V^{-1} e - 2 \frac{q}{\mu} + \frac{q^2}{n\mu} \quad (\text{car } \|v\|^2 = n\mu) \\ &= (V^{-1} e)^T V^{-1} e + \frac{q^2 - 2nq + n^2 - n}{n\mu} - \frac{n^2 - n}{n\mu}. \end{aligned}$$

Or,

$$\begin{aligned} q^2 - 2nq + n^2 - n &= (q - n - \sqrt{n})^2 + 2\sqrt{n}(q - n - \sqrt{n}) \\ &\geq 0 \quad \forall q \geq n + \sqrt{n} \end{aligned}$$

avec égalité si $q = n + \sqrt{n}$.

D'où

$$\frac{q^2}{n^2\mu^2} \|r\|^2 \geq (V^{-1}e)^T V^{-1}e - \frac{n^2 - n}{n\mu}$$

avec égalité si $q = n + \sqrt{n}$ et puisque v est orthogonal à $V^{-1}e - \frac{v}{\mu}$. Alors, en remplaçant $q = n + \sqrt{n}$, et en utilisant la définition de r , on aura :

$$\begin{aligned} (V^{-1}e)^T V^{-1}e - \frac{n^2 - n}{n\mu} &= \left\| V^{-1}e - \frac{n + \sqrt{n}}{n\mu} v \right\|^2 \\ &= \left\| V^{-1}e - \frac{1}{\mu} v \right\|^2 + \left\| \frac{\sqrt{n}}{n\mu} v \right\|^2. \end{aligned}$$

Donc :

$$\begin{aligned} \frac{q^2}{n^2\mu^2} \|r\|^2 &\geq \left(\frac{1}{v_{\min}} - \frac{1}{\mu} v_{\min} \right)^2 + \frac{1}{n\mu^2} v^T v \\ &= \frac{1}{\mu^2} \left\{ \left[\frac{\mu}{v_{\min}} - v_{\min} \right]^2 + \mu \right\} \\ &= \frac{1}{\mu^2} \left\{ \left[\frac{\mu}{2v_{\min}} - v_{\min} \right]^2 + \frac{3}{4} \frac{\mu^2}{v_{\min}^2} \right\} \\ &\geq \frac{3}{4v_{\min}^2}, \end{aligned}$$

et par la suite :

$$\|r\|^2 \geq \frac{\sqrt{3}}{2v_{\min}} \frac{n\mu}{q}. \square$$

Conséquence :

$\|r\|$ n'est pas suffisamment petit $\implies g_1$ est suffisamment négatif \implies une diminution

assez importante de g quand λ devient plus grande que 0.

Théorème 9 Soit $\tau = 0.5$, On définit :

$$\bar{\lambda} = \frac{v_{\min}}{2 \|r\|}.$$

Alors

$$g(0) - g(\bar{\lambda}) \leq -0.15,$$

et donc $\delta = 0.15$.

Preuve

D'après 4.13 on a :

$$\|X^{-1}d_x\|_2 \leq \frac{\|r\|}{v_{\min}},$$

il s'en suit que :

$$\bar{\delta} \|X^{-1}d_x\|_{\infty} \leq \bar{\delta} \|X^{-1}d_x\|_2 \leq \frac{1}{2} \frac{v_{\min}}{\|r\|} \frac{\|r\|}{v_{\min}} = \frac{1}{2} = \tau$$

de la même façon, on vérifie que $\bar{\delta} \|S^{-1}d_s\|_{\infty} \leq \tau$.

D'autre part :

$$\begin{aligned} g(0) - g(\bar{\lambda}) &= \bar{\lambda}g_1 + \frac{1}{2}\bar{\lambda}^2g_2 \\ &\leq -\frac{q}{n\mu}\bar{\lambda}\|r\|^2 + \frac{1}{2}\bar{\lambda}^2\frac{1}{v_{\min}^2}\frac{1}{1-\tau}\|r\|^2 \\ &= -\frac{q}{n\mu}\frac{v_{\min}}{2}\|r\| + \frac{1}{8(1-\tau)}. \end{aligned}$$

En utilisant la borne inférieure de $\|r\|$ et en prenant $\tau = 0.5$, on obtient :

$$g(0) - g(\bar{\lambda}) \leq -\frac{\sqrt{3}}{4} + \frac{1}{8(1-0.5)} = \frac{1-\sqrt{3}}{4} \leq -0.15.$$

Finalement, on aura :

$$\begin{aligned}
\min_{\lambda \in (0, \lambda_{\max})} \Phi_q(x + \lambda d_x, s + \lambda d_s) &\leq \Phi_q(x + \lambda d_x, s + \lambda d_s) \\
&\leq g(\bar{\lambda}) \\
&\leq g(0) - 0.15 \\
&= \Phi_q(x, s) - 0.15.
\end{aligned}$$

donc, la réduction à chaque itération est de 0.15. \square

Remarque 15 *On obtient des résultats similaires pour tout $\tau \in (0, 1)$.*

Corollaire 3 *Soit $q \geq n + \sqrt{n}$ et $\epsilon > 0$.*

*On suppose que le point initial $(x^0, y^0, s^0) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$ et vérifie :*

$$\Phi_q(x^0, s^0) \leq l(q - n) |\log \epsilon|$$

pour un certain $l > 0$ indépendant de n . Alors l'indice K définie par :

$$K = \lceil \frac{l+1}{0.15} (q - n) |\log \epsilon| \rceil = O((q - n) |\log \epsilon|)$$

possède la propriété suivante :

$$(x^k, y^k, s^k) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D, \quad \mu_k \leq \epsilon,$$

pour tout $k \geq K$. En particulier, si $q = n + \sqrt{n}$ alors $K = O(\sqrt{n} \log \epsilon)$.

Preuve

En substituant le majorant de $\Phi_q(x^0, s^0)$ dans la formule 4.10, on aura le résultat. \square

Remarque 16 *En pratique, la réduction δ est plus grande que celle trouvée précédemment.*

CHAPITRE 5

LES MÉTHODES DE LA TRAJECTOIRE CENTRALE

5.1 Introduction

C'est à Barnes que l'on doit le développement en 1987 [8] de cette nouvelle approche des méthodes du point intérieur. La méthode de la trajectoire centrale a été étudiée en premier par Bayer et Lagarias [9] [10] [11] ensuite par Meggido [41], Ben Daya et Shetty [12], Renegar [48], Gonzaga [28], Vaidya [68], Monteiro et Adler [46], Kojima et *al.* [35], Roos et Vial [53][52], Goldfarb et Liu [23], etc.

Comme son nom l'indique, cette méthode est basée sur le suivi d'une trajectoire centrale. Selon les cas, cette trajectoire centrale est définie comme l'ensemble des minimums de la fonction auxiliaire considérée (fonction de pénalité [17], fonction de potentiel [32], fonction des centres [29] ...). Cette méthode consiste à demeurer dans un certain voisinage

de la trajectoire centrale en utilisant des itérations de Newton. Il existe deux approches différentes des méthodes de la trajectoire centrale : une approche primale et une autre primale-duale. Pour mettre en oeuvre chacune de ces approches, nous pouvons utiliser soit des algorithmes à petits pas, soit des algorithmes à grands pas. Tout dépend de la manière de mise à jour des paramètres. Les méthodes primales-duales sont les plus utilisées en pratique. Elles sont efficaces et peuvent s'étendre facilement à d'autres types de problèmes (problèmes quadratiques, problèmes convexes, etc). Le premier algorithme de ce type a été développé par Kojima, Mizuno, et Yoshise [36]. Cet algorithme à grand pas a une complexité de l'ordre de $O(nL)$ itérations. Ces mêmes chercheurs ont décrit un autre algorithme primal-dual mais à petits pas [35], dont la complexité est meilleure et est de l'ordre ($O(\sqrt{n}L)$ itérations seulement). Un algorithme similaire à ce dernier a été proposé par Monteiro et Adler [46]. Après une recherche intense a été faite dans ce domaine et a donné naissance à un nouveau type d'algorithme primal-dual appelé algorithme prédicteur-correcteur. Dans ce chapitre, il sera question tout d'abord des méthodes primales duales. On commence par les décrire, puis on énonce les différents types d'algorithmes, à savoir l'algorithme à petits pas, l'algorithme à grand pas et l'algorithme prédicteur-correcteur. Une deuxième partie sera consacrée à l'approche primale.

5.2 Préliminaires

On considère le problème :

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

et son dual :

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

Soit $\mu > 0$.

À tout problème (P) , on associe le problème pénalisé suivant :

$$(P_\mu) \begin{cases} \min B_\mu(x) \\ Ax = b \end{cases}$$

où B_μ est la fonction de pénalité définie pour $x > 0$ par :

$$B_\mu(x) = c^T x - \mu \sum_{j=1}^n \log x_j$$

Propriétés de B_μ

- Pour tout $\mu > 0$, la fonction B_μ est strictement convexe;
- Si $\overset{\circ}{F}_P$ et $\overset{\circ}{F}_D$ sont non vides, alors Pour tout $\mu > 0$, le problème P_μ admet une solution unique notée $x(\mu)$ et appelée point central;
- Quand $\mu \rightarrow 0$, $x(\mu) \rightarrow x^*$ solution optimale du problème (P) ;
- La fonction $\mu \rightarrow x(\mu)$ définit la trajectoire centrale qu'on note C ;
- $x(\mu)$ est définie d'une façon unique par les conditions d'optimalité de Karush-Kuhn-Tucker suivantes :

$$\begin{cases} c - \mu X^{-1}e - A^T y = 0 \\ Ax = b \end{cases}$$

où y est le vecteur de mutiplicateur de Lagrange associé à la contrainte $Ax = b$ du problème (P_μ) .

En introduisant $s = \mu X^{-1}e$, on aura $xs = \mu e$. Le système précédent devient :

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

Notons que (S_0) correspond aux conditions de complémentarité pour un problème de programmation linéaire.

Remarque 17 *Le système (S_μ) désigne aussi les conditions d'optimalité du problème (D_μ) où*

$$(D_\mu) \begin{cases} \max b^T y + \mu \sum_{j=1}^n \log s_j \\ A^T y + s = c \end{cases}$$

avec $s > 0$.

En effet : les conditions d'optimalité de Karush-Khun-Tucker pour ce dernier problème sont données par :

$$\begin{cases} b - Ax = 0 \\ \mu S^{-1} e - x = 0 \\ A^T y + s = c \end{cases}$$

où x est le vecteur de multiplicateur associé à la contrainte $A^T y + s = c$.

On a : $s > 0$ donc x l'est aussi (car $\mu S^{-1} e - x = 0$).

En ajoutant les deux contraintes $s > 0$ et $x > 0$ au dernier système, on retrouve bien (S_μ) .

5.3 Les méthodes primales duales

Rappelons que les conditions d'optimalité des problèmes (P_μ) et (D_μ) sont données par le système suivant :

$$(S_\mu) \begin{cases} xs = \mu e \\ Ax = b \\ x > 0 \\ A^T y + s = c \\ s > 0 \end{cases}$$

où x désigne la variable primale et (y, s) désignent les variables duales.

5.3.1 Principe

Comme toute méthode de la trajectoire centrale, le but est d'essayer de résoudre le système (S_μ) . En supposant que F_P et F_D sont d'intérieur non vide, on aura alors une solution unique pour le système (S_μ) et ceci pour tout $\mu > 0$. On note cette solution $(x(\mu), s(\mu))$. Dans ce cas, la trajectoire centrale est la courbe de la fonction $\mu \mapsto (x(\mu), s(\mu))$. D'après la première équation du système (S_μ) , $x_i(\mu)s_i(\mu) = \mu$ pour tout $1 \leq i \leq n$. Donc, $x^T s = n\mu$ où $x^T s$ désigne la marge duale; donc, il suffit de trouver (approximativement) des points sur la trajectoire centrale quand $\mu \mapsto 0$. Voilà maintenant un modèle d'algorithme primal-dual de type suivi de la trajectoire.

Modèle général d'algorithme primal-dual

- Données: $\epsilon > 0$, $\mu^0 > 0$
 $k = 0$
- Répéter
 - Trouver approximativement $(x(\mu^k), s(\mu^k))$;
 - Choisir $\mu^{k+1} < \mu^k$.
- $k = k + 1$
- Jusqu'à: $\mu^k < \epsilon$

Remarque 18 1. La résolution de S_μ est approximative donc, il est nécessaire de définir le concept du voisinage du point central.

2. Cet algorithme dépend de la mise à jour du paramètre (choix de μ^{k+1} à chaque itération k).

Concept de proximité

Puisqu'il est impossible de calculer le point central $(x(\mu^k), s(\mu^k))$ d'une façon exacte c'est à dire qui vérifie $\frac{x(\mu^k)s(\mu^k)}{\mu^k} = e$, on se contente d'une solution approchée vérifiant un certain critère de proximité.

Soit $\beta \in (0, 1)$, $k \in \mathbb{N}$ et $(x, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$. La mesure de proximité est définie par :

$$\delta(x, s, \mu) = \left\| \frac{xs}{\mu} - e \right\|_k,$$

et alors, un voisinage de la trajectoire centrale C est défini par :

$$N_k(\beta) = \{(x, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D, \delta(x, s, \mu) \leq \beta\}$$

ou encore

$$N_k(\beta) = \{(x, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D, \|Xs - \mu e\|_k < \beta\mu\}.$$

Les voisinages les plus utilisés dans les méthodes du point intérieur et plus particulièrement dans les méthodes de la trajectoire centrale sont $N_2(\beta)$, $N_\infty(\beta)$ et $N_\infty^-(\beta)$ où $\|m\|_\infty^- = \|m^-\|_\infty$ et $(m^-)_i = \min\{m_i, 0\}$ pour tout vecteur $m \in \mathbb{R}^n$.

Remarque 19

$$C \subset N_2(\beta) \subset N_\infty(\beta) \subset N_\infty^-(\beta) \subset \overset{\circ}{F}_P * \overset{\circ}{F}_D \quad \forall \beta \in (0, 1)$$

Dans ce qui suit on va prendre $k = 2$. (on désigne $\|\cdot\|_2$ par $\|\cdot\|$)

Mise à jour du paramètre

La façon traditionnelle de mettre à jour le paramètre est de le multiplier par un facteur $\sigma \in [0, 1]$ i.e $\mu^{k+1} = \sigma\mu^k$. Ce paramètre σ peut dépendre de n .

La manière de varier ce paramètre à chaque itération entraîne l'obtention de deux catégories de méthodes : les méthodes des petits pas et les méthodes des grands pas.

Maintenant on va résoudre le système (S_μ) . Une façon d'y arriver est de faire une itération de Newton.

Itération de Newton

Soit $(x, s) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$, $\mu > 0$ et $\bar{\mu} = \sigma\mu$. En suivant ce qui précède, notre but est de se rapprocher du point central $(x(\sigma\mu), s(\sigma\mu))$ pour $\sigma \in [0, 1]$. On veut plus précisément trouver $x^+ = x + d_x$ et $s^+ = s + d_s$ tels que (x^+, s^+) réalisable et $x^+s^+ = \bar{\mu}e = \sigma\mu e$. La méthode de Newton résout cela approximativement en linéarisant S_μ . La linéarisation de S_μ donne :

$$\begin{cases} (x + d_x)(s + d_s) = \bar{\mu}e \\ A(x + d_x) = b \\ A^T(y + d_y) + (s + d_s) = c \end{cases}$$

en simplifiant ce système, on aura :

$$(LS_\mu) \begin{cases} sd_x + xd_s = -xs + \bar{\mu}e \\ d_x \in \mathcal{N}(A) \\ d_s \in \mathcal{R}(A^T) \end{cases}$$

qu'on peut représenter par le système suivant :

$$(P_N) \begin{bmatrix} S & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} XSe - \bar{\mu}e \\ 0 \\ 0 \end{bmatrix}$$

où (d_x, d_y, d_s) représente la direction de Newton associée au système (S_μ) (d_x la direction primale et (d_y, d_s) la direction duale).

Par un simple calcul, on obtient :

$$d_x = [S^{-1} - S^{-1}XA^T(AS^{-1}XA^T)^{-1}AS^{-1}](XSe - \bar{\mu}e),$$

$$d_y = -[(AS^{-1}XA^T)^{-1}](XSe - \bar{\mu}e),$$

$$d_s = [A^T(AS^{-1}XA^T)^{-1}AS^{-1}](XSe - \bar{\mu}e).$$

Remarque 20 – Si $\sigma = 0$, on obtient une direction affine qu'on note d_a et donc, le point résultant est : $(x_a, s_a) = (x, s) + d_a$.

– Si $\sigma = 1$, alors ce système définit une nouvelle direction appelée direction centrale notée d_c et alors, $(x_c, s_c) = (x, s) + d_c$.

D'autre part :

$$-xs + \sigma\mu e = \sigma(-xs + \mu e) + (1 - \sigma)(-xs)$$

donc :

$$(d_x, d_y, d_s) = \sigma d_c + (1 - \sigma)d_a.$$

Par conséquent, la direction de Newton associée au système (P_N) est une combinaison convexe d'une direction affine et d'une direction centrale. D'où : $(x^+, s^+) = \sigma(x_c, s_c) + (1 - \sigma)(x_a, s_a)$. De plus, on a : $d_x^T d_s = -d_x^T A^T d_y = 0$, donc la direction qu'on cherche dépend seulement de (x, s) et du paramètre σ . On peut alors écrire $d = (d_x, d_s) = d(x, s, \sigma)$.

Dans ce qui suit, on va montrer que la méthode de Newton est très efficace pour la résolution du système (S_μ) ; en effet, elle réduit la mesure de proximité d'une façon quadratique dans une région assez grande. On prend $\sigma = 1$. (Le cas général où σ est un paramètre quelconque dans $[0, 1]$ sera traité dans la section prochaine (méthode prédicteur-correcteur) avec une certaine condition sur σ et β).

Théorème 10 Soit (x, s) dans $\overset{\circ}{F}_P * \overset{\circ}{F}_D$, $\mu > 0$ tel que $\delta(x, s, \mu) = \delta < 1$ et (x^+, s^+) le résultat de l'itération de Newton pour le système (LS_μ) à partir de (x, s) , alors :

$$\delta(x^+, s^+, \mu) \leq \frac{\delta^2}{\sqrt{8}(1 - \delta)} \quad (5.1)$$

et pour tout $\lambda \in [0, 1]$,

$$\delta(x + \lambda d_x, s + \lambda d_s, \mu) \leq (1 - \lambda)\delta(x, s, \mu) + \lambda^2 \delta(x^+, s^+, \mu). \quad (5.2)$$

De plus, si $\delta \leq 0.7$, alors pour tout $\lambda \in [0, 1]$, le point $((x + \lambda d_x), (s + \lambda d_s))$ est un point intérieur.

La preuve du théorème 10 se sert du lemme suivant.

Lemme 15 Soient $u, v \in \mathbb{R}^n$ tels que $u^T v \geq 0$, alors :

$$\|uv\| \leq \frac{1}{\sqrt{8}} \|u + v\|^2.$$

Preuve

Premièrement, notons que pour tous scalaires ζ, ω vérifiant $\zeta\omega \geq 0$, on a :

$$\sqrt{|\zeta\omega|} \leq \frac{1}{2} |\zeta + \omega|. \quad (5.3)$$

Puisque $u^T v \geq 0$, on a :

$$0 \leq u^T v = \sum_{u_i v_i \geq 0} u_i v_i + \sum_{u_i v_i < 0} u_i v_i = \sum_{i \in \mathcal{K}} |u_i v_i| - \sum_{i \in \mathcal{L}} |u_i v_i|, \quad (5.4)$$

où $\mathcal{K} = \{i/u_i v_i \geq 0\}$, $\mathcal{L} = \{i/u_i v_i < 0\}$ et $\mathcal{K} \cup \mathcal{L} = \{1, 2, \dots, n\}$. Donc,

$$\begin{aligned} \|UVe\| &= (\| [u_i v_i]_{i \in \mathcal{K}} \|^2 + \| [u_i v_i]_{i \in \mathcal{L}} \|^2)^{\frac{1}{2}} \\ &\leq (\| [u_i v_i]_{i \in \mathcal{K}} \|_1^2 + \| [u_i v_i]_{i \in \mathcal{L}} \|_1^2)^{\frac{1}{2}} \quad (\text{puisque } \|\cdot\|_2 \leq \|\cdot\|_1) \\ &\leq (2 \| [u_i v_i]_{i \in \mathcal{K}} \|_1^2)^{\frac{1}{2}} \quad (\text{d'après 5.4}) \\ &\leq \sqrt{2} \left\| \left[\frac{1}{4} (u_i + v_i)^2 \right]_{i \in \mathcal{K}} \right\|_1 \quad (\text{d'après 5.3}) \\ &= 2^{\frac{-3}{2}} \sum_{i \in \mathcal{K}} (u_i + v_i)^2 \\ &\leq 2^{\frac{-3}{2}} \sum_{i=1}^n (u_i + v_i)^2 \\ &\leq 2^{\frac{-3}{2}} \|u + v\|^2, \end{aligned}$$

d'où le résultat. \square

Avant de proposer la preuve du théorème, on considère les notations suivantes : si x et s sont deux vecteurs de même dimension, alors on désigne par xs le vecteur de composantes $x_i s_i$ et $\frac{x}{s}$ le vecteur de composantes $\frac{x_i}{s_i}$.

Preuve du théorème 10

Par un changement de variable du système (LS_μ) , on obtient une forme simple. En effet, soit $h \in \mathbb{R}^n$ un vecteur positif. Considérons le changement de variable suivant :

$$x = h\bar{x}, d_x = h\bar{d}_x, s = h^{-1}\bar{s}, d_s = h^{-1}\bar{d}_s.$$

Notons que $\bar{x}\bar{s} = xs$ et $\bar{d}_x\bar{d}_s = d_xd_s$. Le système (LS_μ) devient :

$$\begin{cases} \bar{x}\bar{d}_s + \bar{s}\bar{d}_x = -\bar{x}\bar{s} + \mu e \\ \bar{d}_x \in \mathcal{N}(\mathcal{A}) \\ \bar{d}_s \in \mathcal{R}(\mathcal{A}). \end{cases}$$

Choisissons $h = \sqrt{\frac{x}{s}}$, il s'ensuit que $\bar{x} = \bar{s} = \sqrt{xs}$, et la première équation du système (LS_μ) peut s'écrire

$$\bar{d}_x + \bar{d}_s = -\frac{\mu}{\bar{x}}\left(\frac{\bar{x}\bar{s}}{\mu} - e\right). \quad (5.5)$$

Or,

$$\left\| \frac{\bar{x}^2}{\mu} - e \right\| = \left\| \frac{xs}{\mu} - e \right\| = \delta$$

et alors, pour $i = 1, \dots, n$, $\left| \frac{\bar{x}_i^2}{\mu} - 1 \right| \leq \delta$. Donc $\frac{\bar{x}_i^2}{\mu} \geq 1 - \delta$ et par conséquent, pour tout $i = 1, \dots, n$,

$$\frac{\mu}{\bar{x}_i^2} \leq \frac{1}{1 - \delta}.$$

En utilisant ce dernier résultat et le résultat 5.4, on en déduit que :

$$\left\| \bar{d}_x + \bar{d}_s \right\|^2 \leq \frac{\mu}{1 - \delta} \left\| \frac{\bar{x}\bar{s}}{\mu} - e \right\|^2 = \mu \frac{\delta^2}{1 - \delta}. \quad (5.6)$$

Maintenant, on établit une borne de complexité le long de la direction (\bar{d}_x, \bar{d}_s) .

On a

$$\begin{aligned} (x + \lambda d_x)(s + \lambda d_s) &= xs + \lambda(xd_s + sd_x) + \lambda^2 d_x d_s \\ &= (1 - \lambda)xs + \lambda\mu e + \lambda^2 d_x d_s. \end{aligned}$$

En utilisant le système LS_μ avec $\sigma = 1$, il résulte que

$$\frac{(x + \lambda d_x)(s + \lambda d_s)}{\mu} - e = (1 - \lambda)\left(\frac{xs}{\mu} - e\right) + \lambda^2 \frac{d_x d_s}{\mu}.$$

En particulier, pour $\lambda = 1$, on a :

$$\delta(x, s, \mu) = \left\| \frac{x^+ s^+}{\mu} - e \right\| = \left\| \frac{d_x d_s}{\mu} \right\|,$$

et pour $\lambda \in [0, 1]$

$$\delta(x + \lambda u, s + \lambda v, \mu) \leq (1 - \lambda)\delta(x, s, \mu) + \lambda^2 \delta(x^+, s^+, \mu).$$

d'où le résultat 5.3. D'autre part, puisque $d_x \in \mathcal{N}(\mathcal{A})$, $d_s \in \mathcal{R}(\mathcal{A}^T)$ et $\bar{d}_x^T \bar{d}_s = d_x^T d_s = 0$ alors par le lemme 16,

$$\|d_x d_s\| = \|\bar{d}_x \bar{d}_s\| \leq \frac{\|\bar{d}_x + \bar{d}_s\|^2}{\sqrt{8}}.$$

En utilisant ce résultat et le résultat 5.5, on déduit que

$$\delta(x^+, s^+, \mu) = \left\| \frac{d_x d_s}{\mu} \right\| \leq \frac{\delta^2}{\sqrt{8}(1 - \delta)}$$

d'où le résultat 5.2. Maintenant on établit la réalisabilité. Puisque (d_x, d_s) est réalisable par construction, il reste à montrer que le ségment $\{(x + \lambda d_x, s + \lambda d_s) / \lambda \in [0, 1]\}$ ne touche pas la frontière de \mathbb{R}_+^n . Calculons la formule 5.2 pour $\delta \leq 0.7$. On obtient $\delta(x^+, s^+, \mu) < 1$, et d'après 5.3, on aura pour tout $\lambda \in [0, 1]$,

$$\delta(x + \lambda d_x, s + \lambda d_s, \mu) < 1.$$

Supposons par l'absurde qu'il existe un certain $\bar{\lambda} \in [0, 1]$ et un certain $i = 1, \dots, n$ tels que $x_i + \bar{\lambda}(d_x)_i = 0$ ou $s_i + \bar{\lambda}(d_s)_i = 0$, on aura

$$\delta(x + \bar{\lambda} d_x, s + \bar{\lambda} d_s, \mu) \geq \left| \frac{(x_i + \bar{\lambda}(d_x)_i)(s_i + \bar{\lambda}(d_s)_i)}{\mu} - 1 \right| = 1.$$

On aboutit à une contradiction avec la dernière inégalité. D'où le résultat est établi. \square

Dans ce qui suit, on présente l'algorithme de la méthode de la trajectoire centrale et ses variantes.

5.3.2 Algorithme primal-dual

1. Initialisation on se donne :

- ϵ un paramètre de tolérance, β, μ_0 ;
- $(x^0, s^0) \in N_2(\beta)$ et $\mu^0 > 0$ tels que $\delta(x^0, s^0, \mu^0) \leq \beta$.

Pour $k = 0, 1, \dots$

2. Répéter :

- Choisir $\sigma \in [0, 1]$.
- Résoudre le système (P_N) avec $(x, s) = (x^k, s^k)$.
- Choisir $\lambda \in (0, 1]$ et soit $(x^{k+1}, s^{k+1}) = (x^k, s^k) + \lambda(d_x, d_s)$.
- Choisir $\mu^{k+1} \in (0, \mu^k)$.
- $k = k + 1$.

3. Jusqu'à $\mu^k < \epsilon$.

À chaque itération, le choix de σ, λ et μ^{k+1} est tel que $\delta(x^k, s^k, \mu^k) \leq \beta$.

Maintenant, on va décrire trois façons intéressantes pour choisir les paramètres.

Algorithme primal-dual à petit pas

À chaque itération, on choisit :

- $\lambda = 1$
- $\mu^{k+1} = \sigma \mu^k$ avec $\sigma = 1 - \frac{0.2}{\sqrt{n}}$.

Proposition 4 *Si on suppose que $\delta(x^0, s^0, \mu^0) \leq 0.5$, alors, pour toute itération k , on a $\delta(x^k, s^k, \mu^k) \leq 0.5$.*

Preuve

Considérons la propriété de récurrence suivante : $\mathcal{P}_k: \delta(x^k, s^k, \mu^k) \leq 0.5$.

Par hypothèse, on a \mathcal{P}_0 . Supposons \mathcal{P}_k et montrons \mathcal{P}_{k+1} . En utilisant le théorème précédent (résultat 5.2), on a

$$\delta(x^{k+1}, s^{k+1}, \mu^k) \leq \frac{0.25}{0.5\sqrt{8}} < 0.2.$$

On doit montrer que :

$$\left\| \frac{x^{k+1}s^{k+1}}{\mu^{k+1}} - e \right\| \leq 0.5.$$

Or, on a :

$$\begin{aligned} \left\| \frac{x^{k+1}s^{k+1}}{\mu^k} - \left(1 - \frac{0.2}{\sqrt{n}}\right)e \right\| &\leq \left\| \frac{x^{k+1}s^{k+1}}{\mu^k} - e \right\| + \left\| \frac{0.2}{\sqrt{n}}e \right\| \\ &= \delta(x^{k+1}, s^{k+1}, \mu^k) + 0.2 \\ &\leq 0.4 \\ &\leq 0.5\left(1 - \frac{0.2}{\sqrt{n}}\right). \end{aligned}$$

D'où \mathcal{P}_{k+1} donc \mathcal{P}_k . L'algorithme primal-dual à petit pas n'est pas très utile dans la pratique car il progresse lentement. Cependant, il donne une preuve immédiate pour la polynomialité de la méthode. \square

Théorème 11 *L'algorithme primal-dual à petit pas avec $\mu^0 \in (0, 2^L)$, $\epsilon = 2^{-2L}$ s'arrête en $O(\sqrt{n}L)$ itérations, avec une paire primal-dual (\hat{x}, \hat{s}) . Une purification à partir de \hat{x} aboutit à un point extrême primal optimal.*

Preuve

Par construction, l'algorithme donne

$$\mu^k \leq \left(1 - \frac{0.2}{\sqrt{n}}\right)^k \mu^0,$$

et par suite

$$\log\left(\frac{\mu^k}{\mu^0}\right) \leq k \log\left(1 - \frac{0.2}{\sqrt{n}}\right) \leq -0.2 \frac{k}{\sqrt{n}}.$$

Or, $\log(1 - x) \leq -x$ pour tout $x \in (0, 1)$, $\mu^0 \in (0, 2^L)$, et pour toute les itérations $\mu^k \geq \epsilon = 2^{-2L}$, il s'ensuit que

$$k \leq -5\sqrt{n} \log \frac{2^{-2L}}{2^L} \leq 15\sqrt{n}L \log 2,$$

d'où $k = O(\sqrt{n})$. Or, $\delta(\hat{x}, \hat{s}, u^k) \leq 0.5$. Donc, en développant cette inégalité on aura :

$$\hat{x}^T \hat{s} \leq (n + 0.5\sqrt{n})\mu^k.$$

En utilisant ce dernier résultat et le critère d'arrêt de l'algorithme, le point résultant satisfait :

$$\hat{x}^T \hat{s} \leq (n + 0.5\sqrt{n})\mu^k < 2n2^{-2L} < 2^{-L}.$$

Puisque $c^T \hat{x} - v^* \leq \hat{x}^T \hat{s}$ alors, une purification à partir de \hat{x} aboutit à la solution optimale de (P). \square

Algorithme primal-dual à grand pas

À chaque itération, on prend

- $\lambda = 1$
- $\mu^{k+1} = \sigma \mu^k$. (On doit déterminer le plus petit réel σ tel que le résultat de l'itération de Newton soit dans le voisinage de la trajectoire centrale fixé d'avance).

Pour cela :

- On résout (P_N) pour $\sigma = 0$ et $\sigma = 1$ et on obtient les paires (x_a, s_a) et (x_c, s_c) .
- On calcule σ tel que $\delta(\sigma x_c + (1 - \sigma)x_a, \sigma s_c + (1 - \sigma)s_a, \sigma \mu) = 0.5$.
- On pose $(x^{k+1}, s^{k+1}) = \sigma(x_c, s_c) + (1 - \sigma)(x_a, s_a)$.

Le calcul de (x_a, s_a) et (x_c, s_c) nécessite la factorisation de deux matrices. Le calcul de σ est un problème à une seule variable et alors, en développant la définition de la mesure de proximité, on montre que σ est une racine d'une équation du 2^e degré et s'il y a plus qu'une racine positive, alors on choisit la plus petite. La borne de polynomialité de $O(\sqrt{n}L)$ est conservée puisqu' à toutes les itérations, σ est plus petit que la valeur utilisée dans l'algorithme à petit pas.

Algorithme prédicteur-correcteur

Il est similaire au précédent, sauf que ses itérations alternent entre deux types de pas : les pas prédictifs et les pas correctifs. On va le voir en détails au paragraphe suivant.

5.3.3 Les méthodes prédictifs-correctifs

Ces méthodes font partie des méthodes primales-duales; elles ont été proposées premièrement par Sonnevend, Stoer et Zhao en 1989 [55], puis étudiées par d'autres. Parmi eux on peut citer : Mizuno, Todd et Ye [44], Zhang [72], Mehrotra [42], Potra [47] ... Ces algorithmes utilisent deux types différents de pas pour atteindre leur objectif. Leurs itérations successives alternent entre ces deux types qui sont les suivants :

- Les pas prédictifs (pour réduire μ);
- les pas correctifs (pour se centrer).

Étude détaillée de la méthode

On considère le problème (P), son dual (D), le domaine réalisable $F_P * F_D$ et son intérieur $\overset{\circ}{F}_P * \overset{\circ}{F}_D$. On note C la trajectoire centrale. Soit $N(\beta)$ un voisinage de C . On a bien

$$C \subset N(\beta) \subset \overset{\circ}{F}_P * \overset{\circ}{F}_D .$$

Soit $(x, s) \in N(\beta)$, la direction $d = (d_x, d_s) = d(x, s, \sigma)$ est donnée par le système (P_N) .

Ayant la direction d , on considère $\theta \in (0, 1]$ et on définit :

$$(P_N)' \begin{cases} x(\theta) &= x + \theta d_x \\ s(\theta) &= s + \theta d_s \end{cases}$$

où θ désigne le pas de l'itération. Les algorithmes de type prédicteur-correcteur utilisent deux voisinages de C tels que l'un est inclu dans l'autre : soit deux réels β_1 et β_2 dans $(0, 1)$ tels que $\beta_1 \leq \beta_2$. On a alors :

$$C \subset N(\beta_1) \subset N(\beta_2) \subset \overset{\circ}{F}_P * \overset{\circ}{F}_D .$$

Pour décrire la méthode, il suffit d'examiner une seule itération. Supposons qu'on commence avec un point $(x^0, s^0) \in N(\beta_1)$ avec $(x^0)^T s^0 \leq 2^t$ (t est un paramètre de tolérance donné d'avance). On calcule la direction du pas prédicteur en posant $\sigma = 0$ dans le système (P_N) . On se déplace le long de cette direction jusqu'à atteindre la frontière de $N(\beta_2)$. On s'arrête à ce point et on définit alors le nouveau point (x^1, s^1) . On calcule ensuite la direction du pas correcteur en prenant $\sigma = 1$ dans le système (P_N) . On se déplace le long de cette direction avec un pas unitaire, on revient alors à $N(\beta_1)$ et on obtient le nouveau point (x^2, s^2) .

Remarque 21 – D'après les systèmes (P_N) et $(P_N)'$ on a :

$$\mu(\theta) = (1 - \theta)\mu + \theta\sigma\mu.$$

Il en résulte que les pas prédicteurs réduisent la valeur de μ par un facteur de $(1 - \theta)$ à chaque itération (car $\sigma = 0$), tandis que les pas correcteurs maintiennent la même valeur de μ (car $\sigma = 1$).

– On choisit $\beta_1 = \frac{1}{4}$ et $\beta_2 = \frac{1}{2}$. Ce choix a aussi été adopté par Sonnevend et al. [55], ainsi que Mizuno et al. [44].

5.3.4 Algorithme prédicteur-correcteur:

1. On se donne un point $(x^0, s^0) \in N(\frac{1}{4})$ vérifiant $(x^0)^T s^0 \leq 2^t$, soit $k = 0$

2. Tant que $(x^k)^T s^k > 2^{-t}$

Pour $k = 0, 1, \dots$

3. Répéter

– Soit $(x, s) = (x^k, s^k)$;

– Pas prédicteur :

– Calculer $d = d(x, s, 0)$ à partir de (P_N) ;

– Calculer le plus grand $\bar{\theta}$ telque

$$(x(\theta), s(\theta)) \in N(\frac{1}{2}) \quad \forall \theta \in [0, \bar{\theta}];$$

– Soit $(x', s') = (x(\bar{\theta}), s(\bar{\theta}))$;

– Pas correcteur :

– Calculer $d' = d(x', s', 1)$ à partir de (P_N) ;

– Soit $(x^{k+1}, s^{k+1}) = (x' + d'_x, s' + d'_s)$;

– $k = k + 1$.

4. Fin.

Remarque 22 *Cet algorithme est similaire à celui de Mizuno et al. [44]. Il diffère de celui de Sonnevend et al. [55] par le fait que ces derniers prennent plusieurs pas correcteurs après chaque pas prédicteur afin d'aboutir à un point suffisamment proche de la trajectoire centrale.*

Lemme 16 On suppose que $(x, s) \in N(\frac{1}{4})$, soit (d_x, d_s) la direction calculée pour le pas prédictif ($\sigma = 0$), alors $(x(\theta), s(\theta)) \in N(\frac{1}{2})$ pour tout $\theta \in [0, \bar{\theta}]$ où :

$$\bar{\theta} = \min\left\{\frac{1}{2}; \sqrt{\frac{\mu}{8 \|D_x D_s e\|}}\right\}$$

Par la suite, le pas prédictif a une longueur au moins égale à $\bar{\theta}$ et la nouvelle valeur de μ est au plus $(1 - \bar{\theta})\mu$.

Pour montrer ce lemme on se sert de quelques résultats techniques.

Résultat 1:

Si $(x, s) \in N(\beta)$, alors $\|D_x D_s e\| \leq \frac{\beta^2 + n(1 - \sigma)^2}{2^{\frac{3}{2}}(1 - \beta)} \mu$ où $D_x = \text{diag}(d_x)$ et $D_s = \text{diag}(d_s)$.

Preuve

On a : $Sd_x + Xd_s = -XSe + \sigma\mu e$. Soit $D = X^{\frac{1}{2}} S^{-\frac{1}{2}}$, en multipliant la première équation par $(XS)^{-\frac{1}{2}}$, on obtient :

$$D^{-1}d_x + Dd_s = (XS)^{-\frac{1}{2}}(-XSe + \sigma\mu e).$$

En appliquant le lemme 16 à $u = D^{-1}d_x$ et $v = Dd_s$, on aura :

$$\begin{aligned} \|D_x D_s e\| &= \|D^{-1}D_x D D_s e\| \\ &\leq 2^{\frac{-3}{2}} \|D^{-1}D_x + D D_s\| \\ &= 2^{\frac{-3}{2}} \|(XS)^{-\frac{1}{2}}(-XSe + \sigma\mu e)\|^2 \\ &= 2^{\frac{-3}{2}} \sum_{i=1}^n \frac{(-x_i s_i + \sigma\mu)^2}{x_i s_i} \\ &\leq 2^{\frac{-3}{2}} \frac{\| -XSe + \sigma\mu e \|^2}{\min_i(x_i s_i)} \end{aligned}$$

et puisque $(x, s) \in N(\beta)$, donc $\min_i(x_i s_i) \geq (1 - \beta)\mu$. D'autre part, on sait que $e^T(XSe - \mu e) = x^T s - \mu e^T e = 0$, donc

$$\begin{aligned} \|XSe - \sigma\mu e\|^2 &= \|XSe - \mu e + (1 - \sigma)\mu e\|^2 \\ &= \|XSe - \mu e\|^2 + \|(1 - \sigma)\mu e\|^2 + 2(1 - \sigma)\mu e^T(XSe - \mu e) \\ &= \|XSe - \mu e\|^2 + (1 - \sigma)^2 \mu^2 n \\ &\leq \beta^2 \mu^2 + (1 - \sigma)^2 \mu^2 n. \end{aligned}$$

D'où :

$$\|D_x D_s e\| \leq \frac{\beta^2 + n(1 - \sigma)^2}{2^{\frac{3}{2}}(1 - \beta)} \mu$$

Ainsi, on obtient le résultat. \square

Résultat 2:

$$X(\theta)S(\theta)e - \mu(\theta)e = (1 - \theta)(XSe - \mu e) + \theta^2 D_x D_s e.$$

Preuve

Soit $i \in \{1, \dots, n\}$, on a :

$$\begin{aligned} x_i(\theta)s_i(\theta) - \mu(\theta) &= (x_i + \theta d_x)(s_i + \theta d_s) - [1 - \theta(1 - \sigma)]\mu \\ &= x_i s_i + \theta(d_x s_i + d_s x_i) + \theta^2 d_x d_s - (1 - \theta + \theta\sigma)\mu \\ &= x_i s_i + \theta(-x_i s_i + \sigma\mu) + \theta^2 d_x d_s - (1 - \theta + \theta\sigma)\mu \\ &= (1 - \theta)x_i s_i + \theta^2 d_x d_s - (1 - \theta)\mu \\ &= (1 - \theta)(x_i s_i - \mu) + \theta^2 d_x d_s. \end{aligned}$$

D'où :

$$X(\theta)S(\theta)e - \mu(\theta)e = (1 - \theta)(XSe - \mu e) + \theta^2 D_x D_s e.$$

D'où le résultat. \square

Résultat 3:

Soit $\beta \in (0, 1)$ et $\sigma \in [0, 1]$ deux paramètres tels que $\frac{\beta^2 + n(1 - \sigma)^2}{2^{\frac{3}{2}}(1 - \beta)} \mu \leq \sigma\beta$.

Alors, si $(x, s) \in N(\beta) \Rightarrow (x(\theta), s(\theta)) \in N(\beta) \quad \forall \theta \in [0, 1]$

Preuve

On a :

$$\begin{aligned}
 \| X(\theta)S(\theta)e - \mu(\theta)e \| &= \| (1 - \theta)(XSe - \mu e) + \theta^2 D_x D_s e \| \\
 &\leq (1 - \theta) \| XSe - \mu e \| + \theta^2 \| D_x D_s e \| \\
 &\leq (1 - \theta)\beta\mu + \theta^2 \frac{\beta^2 + n(1 - \sigma)^2}{2^{\frac{3}{2}}(1 - \beta)} \mu \\
 &\leq (1 - \theta)\beta\mu + \theta^2 \sigma \beta \mu \\
 &\leq (1 - \theta + \sigma\theta)\beta\mu \\
 &= \beta\mu(\theta)
 \end{aligned}$$

donc $(x(\theta), s(\theta))$ satisfait la condition de proximité sur $N(\beta)$.

Il reste à montrer que $(x(\theta), s(\theta)) \in \overset{\circ}{F}_P * \overset{\circ}{F}_D$.

On a bien : $Ax(\theta) = b$ et $A^T y(\theta) + s(\theta) = c$, il reste à montrer que $(x(\theta), s(\theta)) > 0$.

On a $(x, s) = (x(0), s(0)) > 0$.

D'autre part, $x_i(\theta)s_i(\theta) \geq (1 - \beta)\mu(\theta) = (1 - \beta)(1 - \theta(1 - \sigma))\mu$.

Or, $\sigma \in [0, 1]$, $\beta \in (0, 1)$ et $\theta \in (0, 1)$, donc $x_i(\theta)s_i(\theta) > 0$,

et, alors $(x(\theta), s(\theta)) > 0 \quad \forall \theta \in (0, 1)$. D'où le résultat. \square

Preuve du lemme 16

On a :

$$\begin{aligned}
 \| X(\theta)S(\theta)e - \mu(\theta)e \| &\leq (1 - \theta) \| XSe - \mu e \| + \theta^2 \| D_x D_s e \| \\
 &\leq (1 - \theta) \| XSe - \mu e \| + \frac{\mu}{8 \| D_x D_s e \|} \| D_x D_s e \| \\
 &\leq (1 - \theta) \frac{1}{4} \mu + \frac{\mu}{8(1 - \theta)} (1 - \theta) \\
 &\leq \frac{1}{4} (1 - \theta) \mu + \frac{1}{4} (1 - \theta) \mu \quad \text{car } \theta \geq \frac{1}{2} \\
 &\leq \frac{1}{2} (1 - \theta) \mu \\
 &= \frac{1}{2} \mu(\theta).
 \end{aligned}$$

donc $(x(\theta), s(\theta))$ satisfait la condition de proximité pour $N(\frac{1}{2})$.

La vérification de la stricte réalisabilité se fait de la même façon que dans le résultat 3.

□

Conséquence

On peut obtenir une borne inférieure de $\bar{\theta}$. En effet, pour $\beta = \frac{1}{4}$, $\sigma = 0$, on a :

$$\frac{\mu}{8 \| D_x D_s e \|} \geq \frac{2^{1,5}(1 - 0.25)}{8(0.25^2 + n)} = \frac{3\sqrt{2}}{16((0.25^2 + n))} = \frac{3\sqrt{2}}{1 + 16n} > \frac{0.16}{n}$$

donc, $\bar{\theta} \geq \min(\frac{1}{2}, \sqrt{\frac{0.16}{n}}) = \frac{0.4}{\sqrt{n}}$ et, par la suite $\mu^{k+1} \leq (1 - \frac{0.4}{\sqrt{n}})\mu^k$ pour toute itération de l'algorithme.

Lemme 17 Soit $(x, s) \in N(\frac{1}{2})$ et (d_x, d_s) la direction calculée dans la pas correcteur ($\sigma = 1$), alors $(x(1), s(1)) \in N(\frac{1}{4})$ et $\mu(1) = \mu$.

Preuve

Puisque $\mu(\theta) = (1 - \theta(1 - \sigma))\mu$ et $\sigma = 1$ dans le pas correcteur, donc $\mu(1) = \mu$. D'autre part :

$$\begin{aligned} \| X(\theta)S(\theta)e - \mu(\theta)e \| &\leq (1 - \theta) \| XSe - \mu e \| + \theta^2 \| D_x D_s e \| \\ &\leq (1 - \theta)\beta\mu + \theta^2 \frac{\beta^2 + n(1 - \sigma)^2}{2^{\frac{3}{2}}(1 - \beta)}\mu. \end{aligned}$$

En prenant $\beta = \frac{1}{2}$, $\theta = 1$ et $\sigma = 1$, on aura :

$$\| X(1)S(1)e - \mu(1)e \| \leq \frac{\mu}{2^{1,5}\frac{1}{2}} = \frac{\mu}{4\sqrt{2}} \leq \frac{\mu(1)}{4}.$$

D'où $(x(1), s(1))$ satisfait la condition de proximité pour $N(\frac{1}{4})$.

Même chose que le lemme précédent, la stricte réalisabilité se fait de la même façon que le résultat 3. □

Complexité de l'algorithme Prédicteur-correcteur

Ici, on va montrer que la borne de polynomialité de ces méthodes est d'ordre $O(\sqrt{n}L)$ itérations.

Théorème 12 *Étant donné $\epsilon > 0$, on suppose que le point initial $(x^0, s^0) \in N(\frac{1}{4})$ et tel que $\mu^0 \leq \frac{1}{\epsilon^i}$ où i est un entier positif, alors il existe un indice $K = O(\sqrt{n} \log \frac{1}{\epsilon})$ tel que $\mu^k \leq \epsilon \forall k \geq K$.*

Preuve

Pour toute itération k , on a : $\mu^{k+1} \leq (1 - \frac{0.4}{\sqrt{n}})\mu^k$,

donc $\mu^k \leq (1 - \frac{0.4}{\sqrt{n}})^k \mu^0$

ou encore $\log \mu^k \leq k \log(1 - \frac{0.4}{\sqrt{n}}) + \log \mu^0 \leq k \log(1 - \frac{0.4}{\sqrt{n}}) + i \log(\frac{1}{\epsilon})$.

Or, $\log(1 + \alpha) \leq \alpha \forall \alpha > -1$,

donc $\log \mu^k \leq k(-\frac{0.4}{\sqrt{n}}) + i \log(\frac{1}{\epsilon})$.

Or, on veut avoir $\mu^k < \epsilon$, il s'en suit que $k(-\frac{0.4}{\sqrt{n}}) + i \log(\frac{1}{\epsilon}) \leq \log \epsilon$.

Cette inégalité est réalisée pour tout $k \geq K = \frac{\sqrt{n}}{0.4}(1 + i) \log(\frac{1}{\epsilon})$. \square

Extensions

Cette borne de complexité a été obtenu par Mizuno et al. [44], ainsi que par Sonnevend et al. [55]. Ces derniers ont amélioré cette borne de complexité et ont obtenu $O(n^{\frac{1}{4}} \log(\frac{1}{\epsilon}))$ itérations pour des problèmes de programmation linéaire vérifiant $\|D_x D_s e\| \leq \sqrt{n} \mu$ à une itération quelconque de l'algorithme.

Potra a proposé une extension de l'algorithme de Mizuno et al. [44]. Son prédicteur est caractérisé par 2 directions. L'une pour améliorer la complémentarité et l'autre pour améliorer la réalisabilité. La longueur des pas pour les deux directions n'est pas la même,

mais elle est choisie de manière à ce que la complémentarité et la réalisabilité soient améliorées avec le même taux. Le pas correcteur est le même que celui de Mizuno et *al.* [44].

Son algorithme est globalement convergent en supposant que le problème a une solution optimale. Sa complexité est d'ordre $O(n \log(\frac{1}{\epsilon}))$ sous des hypothèses additionnelles sur le point initial. De plus, il exige la factorisation de 2 matrices par itération. Cette extension présente plusieurs inconvénients :

1. La détermination de la longueur du pas exige la résolution d'une équation du 8^e degré.
2. La convergence super linéaire n'a pas été démontrée.

Jiaming Miao [43] a proposé 2 nouveaux algorithmes prédicteurs-correcteurs, où la complémentarité et réalisabilité sont améliorées avec le même taux. L'algorithme 1 est une modification et une extension de l'algorithme de Mizuno et *al.* [44] (version grands pas). Il adopte les mêmes hypothèses proposées par Potra [47] et aboutit au même résultat pour la convergence et la complexité. Cependant, il exige seulement la factorisation de 2 matrices. La longueur du pas est déterminée par la résolution d'une équation quadratique.

Son 2^e algorithme diffère du premier seulement par le choix du pas prédicteur. Sa complexité est aussi d'ordre $O(n \log(\frac{1}{\epsilon}))$.

Récemment, Mehrotra a proposé un algorithme prédicteur correcteur. À chaque itération, il utilise la combinaison de trois directions : direction prédicteur, direction correcteur et une autre direction nommée direction centrale et ceci afin d'améliorer la centralisation. De plus, il exige la factorisation d'une seule matrice.

Jusqu'à présent, son algorithme est considéré comme l'algorithme le plus efficace et le plus utilisé dans la pratique. Il est mis en oeuvre dans une version courante du code OB1

qui est considéré comme l'un des meilleurs codes commercial pour les méthodes du point intérieur.

Vu son importance, il est utile de présenter son approche.

Approche de Mehrotra

Soit $z = (x, y, s)$ et

$$F(z) = \begin{bmatrix} Ax - b \\ A^T y + s - c \\ XSe \end{bmatrix}$$

Les conditions d'optimalité pour les problèmes (P) et (D) sont données par $F(z) = 0$ où $(x, s) > 0$.

Algorithme de Mehrotra:

1. (Initialisation) On se donne un point z^0 tel que $(x^0, s^0) > 0$.

Pour $k = 0, 1, \dots$, répéter

2. Résoudre $F'(z^k)dz = -F(z^k)$ pour dz_p .

3. Résoudre

$$F'(z^k)dz = \begin{pmatrix} 0 \\ 0 \\ dx_p ds_p \end{pmatrix}$$

pour dz_{co} .

4. Choisir μ^k et résoudre $F'(z^k)dz = -\mu^k \hat{e}$ pour dz_{ce} où $\hat{e} = (0, \dots, 0, 1, \dots, 1)$.

5. Soit $dz = dz_p + dz_{co} + dz_{ce}$.

6. Choisir $\tau_k \in [0, 1]$ et soit $\alpha_k = (1, \tau_k \hat{\alpha}_k)$
où $\hat{\alpha}_k = \min\left(\frac{-1}{\min((X^k)^{-1}dx)}, \frac{-1}{\min((S^k)^{-1}ds)}\right)$.

7. $z_{k+1} = z^k + \alpha_k dz$.

Remarque 23 *L'approche de Mehrotra peut s'étendre à d'autres types de problèmes (problèmes convexes, problèmes quadratiques, ...).*

Pour en connaître plus sur l'approche de Mehrotra, voir [42] [57] [73] ...

5.4 Les méthodes primales

On considère le problème (P) ainsi que son problème pénalisé associé (P_μ) .

Rappelons que les conditions d'optimalité de K-K-T pour le problème (P_μ) sont données par :

$$(CKT) \begin{cases} c - \mu X^{-1}e - A^T y = 0 \\ Ax = b \end{cases}$$

où y est le vecteur du mutiplicateur de Lagrange associé à la contrainte $Ax = b$ du problème (P_μ) .

5.4.1 Principe

Comme toute méthode de la trajectoire centrale basée sur les fonctions de pénalité, il y a deux types d'itérations : des itérations principales où on met à jour le paramètre de pénalité et des itérations intermédiaires où on résout les problèmes pénalisés.

En supposant que le domaine réalisable du problème (P) est non vide, alors le système (CKT) admet une seule solution qu'on note $x(\mu)$. Dans ce cas, la trajectoire centrale est donnée par la courbe de la fonction $\mu > 0 \mapsto x(\mu)$.

Comme précédemment, on présente un modèle général d'algorithme primal. Ce dernier diffère de l'algorithme primal-dual par le fait que les itérations sont faites seulement dans l'espace primal.

Modèle général d'algorithme primal

- Données: $\epsilon > 0$, $\mu^0 > 0$, $\sigma \in (0, 1)$.
- $k = 0$
- Répéter
 - Trouver approximativement la solution $x(\mu^k)$ de (P_{μ^k}) .
 - Choisir $\mu^{k+1} = \sigma\mu^k$.
- $k = k + 1$
- Jusqu'à: $\mu^k < \epsilon$

Rappelons que la meilleure façon pour résoudre le problème pénalisé (P_{μ}) est de faire une itération de Newton.

En suivant l'application donnée dans le chapitre 1, la direction de Newton d_N ainsi que y sont données par le système suivant :

$$\begin{pmatrix} \mu X^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} d_N \\ y \end{pmatrix} = \begin{pmatrix} -c + \mu X^{-1}e \\ 0 \end{pmatrix}$$

La matrice μX^{-2} est inversible car $x > 0$ et $\mu > 0$, donc :

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

d'où :

$$\begin{aligned} d_N &= \mu^{-1}X^2[I - A^T(AX^2A^T)^{-1}AX^2](-c + \mu X^{-1}e) \\ &= -X(I - XA^T(AX^2A^T)^{-1}AX)\left(\frac{Xc}{\mu} - e\right) \\ &= -XP_{N(Ax)}\left(\frac{Xc}{\mu} - e\right) \end{aligned}$$

et

$$\begin{aligned}y &= [(AX^2A^T)^{-1}AX^2](c - \mu X^{-1}e) \\ &= (AX^2A^T)^{-1}AX(Xc - \mu e).\end{aligned}$$

De la même façon, puisque la résolution de (P_μ) est approximative, il est nécessaire de définir la notion de mesure de proximité ainsi que la notion de voisinage de la trajectoire centrale dans le cas primal.

Définitions

Soit $\mu > 0$, $x \in \overset{\circ}{F}_P$, $\beta \in (0, 1)$.

La mesure de proximité de x à $x(\mu)$ est définie par :

$$\delta(x, \mu) = \| X^{-1}d_N \|_k,$$

et alors, un voisinage de la trajectoire centrale est défini par :

$$N(\beta) = \{x \in \overset{\circ}{F}_P, \delta(x, \mu) \leq \beta\}$$

Remarque 24 Dans le cas primal, on prend toujours $k = 2$.

Notons que si $\delta(x, \mu) < 1$, alors l'itération de Newton donne un nouveau point intérieur.

Proposition 5 En suivant les notations précédentes, si on pose $s = c - A^T y$ alors,

$$\delta(x, \mu) = e - \frac{Xs}{\mu}$$

Preuve

$$\begin{aligned}\delta(x, \mu) &= X^{-1}d_N \\ &= -(I - XA^T(A X^2 A^T)^{-1}AX)\left(\frac{Xc}{\mu} - e\right) \\ &= \left(e - \frac{1}{\mu}Xc\right) + XA^T(A X^2 A^T)^{-1}AX\left(\frac{1}{\mu}Xc - e\right).\end{aligned}$$

D'autre part:

$$\begin{aligned}e - \frac{Xs}{\mu} &= e - \frac{X}{\mu}(c - A^T y) \\ &= \left(e - \frac{1}{\mu}Xc\right) + XA^T \frac{1}{\mu}y \\ &= \left(e - \frac{1}{\mu}Xc\right) + XA^T(A X^2 A^T)^{-1}AX\left(\frac{1}{\mu}Xc - e\right).\end{aligned}$$

d'où le résultat. \square

Remarque 25 y et s désignent respectivement la variable duale et la variable d'écart du problème (D).

Mise à jour du paramètre

Comme pour l'approche primal-dual, on distingue deux types de méthodes :

- Les méthodes des petits pas ont été développées premièrement par Gill et *al.* [22] puis par Gonzaga [24], Renegar [48], Roos [50],[51], Roos et Den Hertog [49] et Roos et Vial [53]... Dans ces méthodes, on choisit σ assez petit, donc la variation du paramètre est presque négligeable. Alors, on progresse lentement tout en restant très proche de la trajectoire centrale (une seule itération de Newton suffira pour

obtenir la nouvelle itération). La borne de polynomialité de ces méthodes est d'ordre $O(\sqrt{n}L)$ itérations. Cependant, elles sont peu utiles dans la pratique. L'étude de la complexité de ces algorithmes est semblable à celle des algorithmes primals-duals à petit pas.

- Les méthodes des grands pas ont été développées premièrement par Roos et Vial [52] et Gonzaga [26]. Dans ces méthodes, la réduction du paramètre est assez importante, donc les itérations obtenues lors de la résolution des problèmes pénalisés peuvent être hors du voisinage de C et plusieurs itérations de Newton sont nécessaires pour retourner au champs de la trajectoire centrale. Malgré que la borne de polynomialité de ces méthodes est d'ordre $O(nL)$ itérations, elles sont très utiles dans la pratique.

Il est donc intéressant de proposer un algorithme primal à grand pas et d'étudier sa complexité.

Pour faciliter notre étude, on prend comme fonction de pénalité

$$f(x, \mu) = \frac{c^T x - v^*}{\mu} - \sum_{j=1}^n \log x_j$$

où v^* est la valeur optimale du problème (P) .

On suppose qu'on a :

- Un point initial x^0 strictement réalisable et vérifiant $x_j^0 \geq 2^{-L}$,
- Une borne inférieure v_L de la valeur optimale du problème (P) tel que $|v_L|$ est d'ordre $O(2^L)$.

L'algorithme à grand pas est le suivant :

5.4.2 Algorithme à grand pas

1. On se donne :

- x^0 un point réalisable pour (P);
- v_L une borne inférieure de la valeur optimale;
- t un paramètre de tolérance.

2. Soit :

$$\mu_0 = c^T x^0 - v_L; x = x^0; \mu = \mu_0$$

Tant que : $\mu > 2^{-t}$ répéter

Itération principale :

- Tant que $\| \delta(x, \mu) \| \geq 1$, répéter
- Itération intermédiaire :
 - $\bar{\alpha} = \arg \min_{\alpha > 0} \{ f(x + \alpha X \delta(x, \mu), \mu) : x + \alpha X \delta(x, \mu) > 0 \}$
 - $x = x + \bar{\alpha} X \delta(x, \mu)$
- Fin de l'itération intermédiaire :
- $\mu = \frac{\mu}{2}$

Fin de l'itération principale :

3. Fin.

Remarque 26 *Le facteur de réduction de μ est arbitraire. La valeur $\frac{1}{2}$ peut être remplacée par n'importe quelle valeur dans $(0,1)$.*

5.4.3 Complexité de l'algorithme

Dans ce paragraphe on va montrer que l'algorithme converge en $O(nL)$ itérations. Cependant on se sert de quelques lemmes.

Lemme 18 Si $\|\delta(x, \mu)\| \leq 1$ alors $c^T x - v^* \leq \mu(n + \sqrt{n})$.

Preuve

D'après la définition de $\delta(x, \mu)$, on a :

$$\left\| e - \frac{Xs}{\mu} \right\| \leq 1$$

ceci implique que $s \geq 0$. Donc, y est une variable duale réalisable et par la suite $b^T y \leq v^*$.

D'autre part,

$$\left| e^T \left(\frac{Xs}{\mu} - e \right) \right| \leq \|e\| \left\| \frac{Xs}{\mu} - e \right\| \leq \sqrt{n}$$

et

$$e^T \left(\frac{Xs}{\mu} - e \right) = \frac{x^T s}{\mu} - n = \frac{c^T x - b^T y}{\mu} - n \geq \frac{c^T x - v^*}{\mu} - n.$$

D'où le résultat. \square

Lemme 19 Si x est un point tel que $\|\delta(x, \mu)\| \leq 1$ et $0 < \theta < 1$, alors :

$$f(x, \theta\mu) \leq f(x, \mu) + \frac{1 - \theta}{\theta} (n + \sqrt{n})$$

Preuve

en utilisant le lemme suivant, on obtient directement le résultat. En effet :

$$\begin{aligned} f(x, \theta\mu) &= \frac{c^T x - v^*}{\theta\mu} - \sum_{j=1}^n \log x_j \\ &= \frac{1 - \theta}{\theta} + \frac{c^T x - v^*}{\mu} - \sum_{j=1}^n \log x_j \\ &\leq \frac{1 - \theta}{\theta} (n + \sqrt{n}) + f(x, \mu). \end{aligned}$$

Considérons le cas où la longueur du pas choisi au point x le long de la direction d_N est α . En notant Δf la variation de la fonction de barrière, il en résulte que :

$$-\Delta f = -\alpha \frac{c^T X \delta(x, \mu)}{\mu} + \sum_{j=1}^n \log(1 + \alpha \delta_j(x, \mu)).$$

D'où le résultat. \square

Lemme 20 Soit $\alpha = (1 + \|\delta(x, \mu)\|_\infty)^{-1}$. Alors :

$$-\Delta f \geq \|\delta(x, \mu)\| - \log(1 + \|\delta(x, \mu)\|).$$

Preuve

Voir[52]. \square

Lemme 21 L'algorithme s'arrête après K itérations principales où K est le plus petit entier vérifiant :

$$K \geq t + \log \mu^0$$

Preuve

Soit $\{\mu_k\}$, $k = 1, \dots, K$ la suite des valeurs de μ engendrée par des itérations principales successives. Alors, $\mu_k = 2^{-k} \mu_0$. Or, l'algorithme s'arrête après K itérations principales lorsque $2^{-K} \mu_0 < 2^{-t}$. Ceci implique le résultat. \square

Pour s'assurer que la solution à laquelle l'algorithme s'arrête soit utilisée pour calculer une solution de base dans un temps polynomial, il suffit de prendre $t = O(L)$.

Théorème 13 L'algorithme s'arrête en au plus $4[(t + \log u_0)(n + \sqrt{n}) + 1 + 2nL]$ itérations intermédiaires. Le dernier point engendré par l'algorithme satisfait :

$$c^T x - v^* \leq 2^{-t}(n + \sqrt{n}).$$

Preuve

Soit x^k le point obtenu à la fin de la k^e itération. Donc, en utilisant le lemme 20 avec $\theta = \frac{1}{2}$, on aura :

$$f(x^k, \mu_k) \leq f(x^k, \mu_{k-1}) + n + \sqrt{n}.$$

D'autre part, d'après le lemme 21, la diminution de la fonction f pendant l'itération intermédiaire satisfait

$$-\Delta f \geq \|\delta(x, \mu)\| - \log(1 + \|\delta(x, \mu)\|).$$

Or, $\|\delta(x, \mu)\| - \log(1 + \|\delta(x, \mu)\|)$ est une fonction croissante en $\|\delta(x, \mu)\|$, de plus durant l'itération intermédiaire $\|\delta(x, \mu)\| \geq 1$, donc $-\Delta f \geq 1 - \log 2 > \frac{1}{4}$.

D'où :

$$f(x^k, \mu_k) \leq n + \sqrt{n} + f(x^{k-1}, \mu_{k-1}) - \frac{1}{4}p_k,$$

où p_k désigne le nombre d'itérations intermédiaires pendant la k^e itération principale. Il en résulte que :

$$f(x^k, \mu_k) \leq k(n + \sqrt{n}) + f(x^0, \mu_0) - \frac{1}{4} \sum_{j=1}^k p_j.$$

Ainsi, on obtient :

$$\frac{1}{4} \sum_{j=1}^K p_j \leq K(n + \sqrt{n}) + f(x^0, \mu_0) - f(x^K, \mu_K).$$

En notant N le nombre total des itérations intermédiaires, on aura :

$$\frac{1}{4}N \leq K(n + \sqrt{n}) + f(x^0, \mu_0) - f(x^K, \mu_K).$$

D'autre part, d'après la définition de μ^0 et la supposition 2 on a :

$$f(x^0, \mu^0) = 1 - \sum_{j=1}^n \log x_j^0 \leq 1 + nL.$$

de plus, puisque le point x^K est réalisable pour le problème (P), il peut s'écrire comme combinaison convexe des solutions de base réalisables. Or, la j^e coordonnée de chaque

solution de base réalisable satisfait : $x_j \leq 2^L$, $j = 1, \dots, n$. Donc :

$$\sum_{j=1}^n \log x_j^K \leq nL.$$

Et puisque $c^T x^K - v^*$ est positive, il s'en suit que :

$$-f(x^K, \mu_K) = \sum_{j=1}^n \log x_j^K - \frac{c^T x^K - v^*}{\mu - K} \leq nL.$$

d'où :

$$\frac{1}{4}N \leq K(n + \sqrt{n}) + 1 + nL + nL.$$

ou encore :

$$N \leq 4[K(n + \sqrt{n}) + 1 + 2nL].$$

Or $K \geq t + \log \mu^0$. D'où le résultat.

Le deuxième résultat se déduit directement du lemme 18. \square

Remarque 27 Si on réduit μ par un facteur θ , $0 < \theta < 1$ (au lieu de $\frac{1}{2}$), alors dans la dernière inégalité de la preuve précédente le terme $K(n + \sqrt{n})$ sera multiplié par $\frac{1 - \theta}{\theta}$. En conséquence, pour trouver la solution optimale du problème (P), l'algorithme exige un $O(L)$ itérations principales (puisque $K = O(L)$). Or d'après la preuve du théorème, le nombre total des itérations intermédiaires est de $O(nL)$ itérations. Donc la complexité totale de l'algorithme à grands pas pour la méthode de la trajectoire centrale est de $O(nL)$ itérations. Cette borne de complexité a aussi été obtenue par Gonzaga [26], mais en utilisant une autre fonction auxiliaire.

CHAPITRE 6

CONCLUSION

Dans ce travail, nous avons présenté une synthèse sur les différentes méthodes de point intérieur à savoir les méthodes affines, les méthodes projectives, les méthodes de potentiel et les méthodes de la trajectoire centrale. Ainsi, nous avons exposé les principes de ces méthodes et présenté certains algorithmes de résolution pour les mettre en oeuvre. Nous avons également étudié la complexité de certains algorithmes. En résumé, les tableaux 1 et 2 présentent les similarités et différences entre les méthodes étudiées.

Donc, dans tous les cas les directions cherchées sont des combinaisons linéaires d'une direction affine et d'une direction centrale ($X^k P_{AX^k e}$) (même pour les méthodes prédicteurs-correcteurs). De plus, les conditionnements utilisés sont soit avec X^k dans le cas primal, soit avec $(S^k)^{-1}$ dans le cas dual ou $(X^k (S^k)^{-1})^{0.5}$ dans le cas primal-dual.

En se basant sur les mises en oeuvre de ces méthodes et en tenant compte de leurs complexités, il ressort que les méthodes de type primal-dual, sont les plus efficaces dans la pratique et plus particulièrement les méthodes de type prédicteur-correcteur.

Méthodes du point intérieur	Méthodes affines	Méthodes projectives	Méthodes du potentiel	Méthodes de la trajectoire centrale
Problème étudié	P (p.3)	P_1 (p.32)	P	P
Transformation utilisée	affine $x \mapsto (X^k)^{-1}x$	projective $x \xrightarrow{T_P} \hat{x} = \frac{n(X^k)^{-1}x}{e^T(X^k)^{-1}x}$	affine $x \mapsto (X^k)^{-1}x$	
Fonction auxiliaire		$n \log c^T x - \sum_{j=1}^n \log x_j$	$q \log c^T x - \sum_{j=1}^n \log x_j$	$c^T x - \mu \sum_{j=1}^n \log x_j$
Direction	$d_{aff} = -X^k P_{AX^k} X^k c$	$d_{kar} = -X^k P_{\hat{B}^k} X^k c$	$d_{pot} = -X^k P_{AX^k} \nabla \bar{f}(e)$	$d = \frac{1}{\mu} d_{aff} - X^k P_{AX^k} e$
Complexité		$O(n^{3.5}L)$	$O(n^{3.5}L)$	$O(n^{3.5}L)$

TAB. 6.1: *Cas primal*

Méthodes du point intérieur	Méthodes du potentiel	Méthodes de la trajectoire centrale
Conditionnement	$D = (X^k(S^k)^{-1})^{0.5}$	$D = (X^k(S^k)^{-1})^{0.5}$
Direction utilisée	de Newton associée à S_μ $\bar{\mu} = \sigma\mu, \sigma = \frac{q}{n}, q = O(n),$	de Newton associée à S_μ $\bar{\mu} = \sigma\mu, \sigma \in [0, 1]$
Complexité	$O(n^{3.5}L)$ si $q \geq n + \sqrt{n}$	$O(n^{3.5}L)$
Meilleure complexité	$O(n^3L)$	$O(n^3L)$

TAB. 6.2: *Cas primal-dual*

Bibliographie

- [1] I. Adler, N. Karmarkar, M. Resende, and G. Veiga. Data structures and programming techniques for implementation of Karmarkar's algorithm, *ORSA Journal on Computing*, 1(2):84–106, 1989.
- [2] I. Adler, N. Karmarkar, M. Resende, and G. Veiga. An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44:297–335, 1989.
- [3] K.M. Anstreicher and R.A. Bosch. Long steps in an $O(n^3L)$ algorithm for linear programming. *Mathematical Programming*, 54:251–265, 1992.
- [4] K.M. Anstreicher. A monotonic projective algorithm for fractional linear programming. *Algorithmica*, 1:483–498, 1985.
- [5] K.M. Anstreicher. A standard form variant and safeguarded linesearch for the modified Karmarkar algorithm. *Mathematical Programming*, 47:337–351, 1990.
- [6] K.M. Anstreicher. A combined phase I-phase II scaled potential algorithm for linear programming. *Mathematical Programming*, 52:429–439, 1991.
- [7] E.R. Barnes. A variation of Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36:174–182, 1986.

- [8] E.R. Barnes. A polynomial time version of the affine scaling algorithm. *Presented at the Joint National ORSA/TIMSW Meeting, St.Louis, Octobre 1987.*
- [9] D. Bayer and J.C. Lagarias. The non linear geometry of linear Programming : I. affine and projective scaling trajectories; II. legendre Transform Coordinates; III. central Trajectories *Technical report, AT and T Bell laboratories, Murray Hill, N.J., 1986.*
- [10] D. Bayer and J.C. Lagarias. The nonlinear geometry of linear programming. I affine and projective scaling trajectories *Transactions of the American Mathematical Society*, 314(2):499–526, 1989.
- [11] D. Bayer and J.C. Lagarias. The nonlinear geometry of linear programming. II legendre transform coordinates and central trajectories *Transactions of the American Mathematical Society*, 314(2):527–581, 1989.
- [12] M. Ben Daya and C.M. Shetty. Polynomial barrier function algorithms for linear programming. *Research report 30332-0205, School of industrial and systems engineering, Georgia institute of technology, Atlanta, GA, 1988.*
- [13] D. Bertsimas and J.N. Tsitsiklis. Introduction to Linear Optimization. *Athena Scientific*, 1997.
- [14] T.M. Cavalier and A.L. Soyster. Some computational experience and a modification of Karmarkar algorithm. *Presented at the 12th International Symposium on Mathematical Programming, Massachusetts Institute of Technology, Cambridge, Massachsets, August 1985*
- [15] I.I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics Doklady*, 8:674–675, 1967.
- [16] I.I. Dikin. On the speed of an iterative process. *Upravlyaemye Sistemi*, 12:54–60, 1974.

- [17] A.V. Fiacco and McCormick. Nonlinear programming: Sequential unconstrained minimisation techniques. Technical report, John Wiley, New York, 1968.
- [18] C. Fraley. Linear updates for a single-phase projective method. *Operations Research Letters*, 9:169–174, 1990.
- [19] R.M. Freund. Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical Programming*, 51:203–222, 1991.
- [20] R.M. Freund. A potential-function reduction algorithm for solving linear program directly from an infeasible warm start. *Mathematical Programming*, 52:441–466, 1991.
- [21] D.M. Gay. A variant of Karmarkar’s linear programming algorithm for problems in standard form. *Mathematical Programming*, 37:81–90, 1987.
- [22] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin, and M.H. Wright. On projected newton barrier methods for linear programming and an equivalence to Karmarkar’s projective method. *Mathematical Programming*, 36:183–209, 1986.
- [23] D. Goldfarb and S. Liu. An $O(n^3L)$ primal interior point algorithm for convex quadratic programming. *Mathematical Programming*, 49:325–340, 1990/91.
- [24] C.C. Gonzaga. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In N. Meggido, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 1–28. Springer Verlag New York, 1987.
- [25] C.C. Gonzaga. Polynomial affine algorithm for linear programming. *Mathematical Programming*, 49:7–21, 1990.
- [26] C.C. Gonzaga. Large-step path-following methods for linear programming, part I: barrier function method. *SIAM Journal on Optimization*, 1:280–292, 1991.

- [27] C.C. Gonzaga. Large-step path-following methods for linear programming, part II: potential reduction method. *SIAM Journal on Optimization*, 1:280–292, 1991.
- [28] C.C. Gonzaga. Path following method for linear programming. *SIAM Review*, 34(2):167–224, 1992.
- [29] P. Huard. Resolution of mathematical programming with non linear constraints by the method of centers. In Abadie, editor, *in Nonlinear Programming*. North-Holland, Amsterdam, 1967.
- [30] M. Iri and H. Imai. A multiplicative barrier function method for linear programming. *Algorithmica*, 1:455–482, 1986.
- [31] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1972.
- [32] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceeding of the 16th Annual ACM. Symposium on theory of computing*, pages 303–311, 1984.
- [33] L.G. Khachiyan. A polynomial algorithm for linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
- [34] V. Klee and G.J. Minty. How good is the simplex algorithm. *in O.Shisha (ed.), Inequalities III Academic, New York*, pages 159–175, 1974.
- [35] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44:1–26, 1989.
- [36] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. In N. Meggido, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 29–47. *Springer Verlag New York*, 1989.

- [37] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for complementarity problems. *Mathematical Programming*, 50:331–342, 1991.
- [38] K.O. Kortanek and M. Shi. Convergence results and numerical experiments on a linear programming hybrid algorithm. *European Journal of Operational Research*, 32:47–61, 1987.
- [39] J.C. Lagarias. The nonlinear geometry of linear programming. III projective legendre transform coordinates and hilbert geometry. *Transactions of the American Mathematical Society*, 320(1):193–225, 1990.
- [40] K.A. McShane, C.L. Monma, and D.F. Shanno. An implementation of a primal-dual interior point method for linear programming. *ORSA journal on computing*, 1:70–83, 1989.
- [41] N. Meggido. Introduction: New approaches to linear programming. *Algorithmica*, 1(4):387–394, 1986.
- [42] S. Mehrotra. On the implementation of a primal-dual interior-point method. *SIAM Journal of Optimization*, 2(4):208–227, 1992.
- [43] J. Miao. Two infeasible-interior point predictor-corrector algorithms for linear programming. *SIAM Journal of Optimization*, 6(3):587–599, 1996.
- [44] S. Mizuno, M.J. Todd, and Y. Ye. On adaptive-step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18(4):964–981, 1993.
- [45] C.L. Monma and A.J. Morton. Computational experience with a dual affine variant of Karmarkar’s method for linear programming. *Operations Research Letters*, 6:443–482, 1993.

- [46] R.D.C. Monteiro and I. Adler. Interior path following primal-dual algorithms. part I: Linear programming. *Mathematical Programming*, 44:27–41, 1989.
- [47] F.A. Potra. An infeasible-interior point predictor-corrector algorithm for linear programming. *SIAM Journal of Optimization*, 6(1):19–32, 1996.
- [48] J. Renegar. A polynomial-time algorithm based on Newton’s method for linear programming. *Mathematical Programming*, 40:59–93, 1988.
- [49] C. Roos and D.Den. Hertog. A polynomial method of approximate weighted centers for linear programming. *Technical Report 89-13, Faculty of Mathematics and informatics/Computer Science, Delft University of Technology (Delft, Netherlands)*, 1989.
- [50] C. Roos. A new trajectory following polynomial-time algorithm for linear programming. *Journal on optimization theory and its application*, 63(3):433–458, 1987.
- [51] C. Roos. An $O(n^3L)$ approximate center method for linear programming. In S.Dolecki, editor, *Proceeding of the Fifth French-German Conference.lecture notes in Mathematics.No.1405*, pages 147–158. *Springer, Berlin*, 1989.
- [52] C. Roos and J.-Ph. Vial. Long steps with the logarithmic penalty barrier function in linear programming. In *J.Richard et L.Wolsey, editor, lecture notes control inform. Sci.84*, pages 433–441. *Economic Decision-Making: Games, Economics and Optimization (dedicated to Jacques H.Dreze)*, Elsevier science publisher B.V., 1990.
- [53] C. Roos and J.-Ph. Vial. A polynomial method of approximate centers for linear programming. *Mathematical Programming*, 54:295–305, 1992.
- [54] D.F. Shanno and R.E. Marsten. On implementing Karmarkar’s method. *Working Paper 85-1, Graduate School of Administration, University of California at Davis, California*, 1985.

- [55] G. Sonnevend, J. Stoer, and G. Zhao. On the complexity of following the central path of linear programs by linear extrapolation. *Methods of Operations Research*, 63:19–31, 1989.
- [56] K. Tanabe. Centered Newton method for linear programming and quadratic programming. *In Proceedings of the 8th mathematical programming symposium Japan*, pages 131–152, 1987.
- [57] R. Tapia, Y. Zhang, M. Saltzman, and A. Weiser. The Mehrotra predictor-corrector interior-point method as a perturbed composite newton method. *SIAM Journal of Optimization*, 6(1):47–56, 1996.
- [58] T. Terlaky and T. Tsychiya. A note on Mascarenhas' counter example about global convergence of the affine scaling algorithm. *Technical report, University of Iowa*, 1996.
- [59] M.J. Todd and B.P. Burrell. An extension of Karmarkar's algorithm for linear programming using dual variables. *Algorithmica*, 1:409–424, 1986.
- [60] M.J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 2:198–209, 1990.
- [61] M.J. Todd and Y. Ye. Containing and shrinking ellipsoids in the path-following algorithm. *Mathematical Programming*, 47:1–9, 1990.
- [62] M.J. Todd. Combining phase I and phase II in a potential reduction algorithm for linear programming. *Mathematical Programming*, 59:133–150, 1993.
- [63] J.A. Tomlin. An experimental approach to Karmarkar's projective method for linear programming. *Ketron Incorporated*, 1985.
- [64] P. Tseng and Z.Q. Luo. On the convergence of the affine-scaling algorithm. *Mathematical Programming*, 56:301–319, 1992.

- [65] T. Tsychiya. Global convergence property of the affine scaling methods for primal degenerate linear programming problems. *Mathematics of Operations Research*, 17:527–557, 1992.
- [66] T. Tsychiya. Quadratic convergence of Iri and Imai’s algorithm for degenerate linear programming problems. *Journal of Optimisation Theory and Applications*, 87(3):703–726, 1995.
- [67] T. Tsychiya and M. Muramatsu. Global convergence of a long-step affine scaling algorithm for degenerate linear programming problems. *SIAM Journal on Optimization*, 5(3):525–557, 1995.
- [68] P.M. Vaidya. An algorithm for linear programming which require $O((m+n)n^2 + (m+n)^{1.5}nL)$ arithmetic operations. *Mathematical Programming*, 47:175–201, 1990.
- [69] R.J. Vanderbei, M.S. Meketon, and B.A. Freedman. A modification of Karmarkar’s linear programming algorithm. *Algorithmica*, 4(1):395–408, 1986.
- [70] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.
- [71] Y. Ye and M. Kojima. Recovering optimal basis in Karmarkar’s polynomial algorithm for linear programming. *Mathematical Programming*, 39:305–317, 1987.
- [72] Y. Zhang. On the convergence of a class of infeasible interior point methods for the horizontal linear complementarity problem. *SIAM Journal of Optimization*, 4:208–227, 1994.
- [73] Y. Zhang and D. Zhang. On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. *Mathematical Programming*, 68:303–318, 1995.