

الجمهورية الجزائرية الديمقراطية الشعبية
الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire de Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de licence
en: - Filière mathématiques fondamentales**

Thème Méthode de karmarkar

Préparé par :

- Fedala Amina
- Rouba Meriem
- Bourekba Feyrouz

Encadré par: Baiche Kanzia

Année universitaire :2013/2014



Remerciement




Nous avons abouti un travail, qui a été le résultat d'un cheminement de tout un parcours pédagogique, qui a duré tout le long de notre parcours éducatif dans l'enseignement supérieur.

*Un remerciement particulier à notre encadreur **Mme Baiche Kanzia** pour sa présence, son aide et surtout pour ses précieux conseils qui nous ont assistés pour l'accomplissement de notre projet.*

Nous tenons à exprimer nos sincères remerciements à tout le personnel de l'institut des sciences et de la technologie surtout les enseignants qui nous ont formé durant toutes nos années d'étude.

*Nous remercions à l'avance **Mlle Maamar Radja** pour leurs aides, conseils, et surtout pour leurs attentions et suivi.*



Un remerciement particulier à nos très chers parents, frères, sœurs, collègues et amies respectives qui nous ont encouragés, soutenu durant tout notre parcours.

MERCI À TOUS



Rouba Meriem



Bourekba Feyrouz

Fedala Amina



Table des matières

Introduction	3
1 optimisation linéaire sans constraints et avec constraints	5
1.1 Introduction	5
1.2 Notion de base	5
1.2.1 Element d'analyse convexe	5
1.2.2 Rappels d'optimisation	7
1.3 Optimisation sans contraintes	7
1.3.1 Résultats d'existence et d'unicité	7
1.3.2 Conditions d'optimalité	9
1.4 Optimisation avec contraintes	10
1.4.1 Résultats d'existence et d'unicité	10
1.4.2 Condition d'optimalité	11
1.5 Classification de problèmes d'optimisation	13
2 Programmation linéaire	15
2.1 Introduction	15
2.2 Formes générales d'un programme linéaire	15
2.2.1 Forme canonique	15
2.2.2 Forme standard	16

2.3	Equivalence de deux programmes mathématiques	16
2.4	dualité	17
2.4.1	La dualité en programmation linéaire	17
2.4.2	Application de la dualité	19
3	Méthode de karemarkare	20
3.1	Introduction	20
3.2	Préliminaires	20
3.3	L'idée générale de l'algorithme	21
3.4	Programme linéaire traité par Karmarkar	23
3.5	Transformation projective de Karmarkar	25
3.6	Description de l'algorithme de base	26
3.6.1	Algorithme de base	27
3.6.2	Dérivation et analyse de l'algorithme	28
3.7	Convergence de l'algorithme	30
3.8	Généralisation de l'algorithme de Karmarkar	31
3.9	Extension de la méthode de Karmarkar	33
3.9.1	Formulation du problème	33
3.9.2	Linéarisation	35
3.9.3	Description de l'algorithme	35
3.9.4	Etude de la convergence	36
3.9.5	Fonction potentielle et convergence	37
3.10	Calcul d'une solution initiale strictement réalisable	38
	conclusion	40
	Bibliographie	41

Introduction

L'optimisation est une branche des mathématiques, cherchant à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à déterminer le meilleur élément d'un ensemble, au sens d'un critère quantitatif donné. Ce mot vient du latin *optimus* qui signifie le meilleur.

L'optimisation joue un rôle important en recherche opérationnelle (donc en économie et microéconomie). Dans les mathématiques appliquées (fondamentales pour l'industrie et l'ingénierie), en analyse et en analyse numérique, en statistique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande.

Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et de prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique.

En 1984 Narendra Karmarkar se base sur les méthodes de points intérieurs et découvre un algorithme en temps polynomial. Son travail donne un véritable élan à la recherche d'algorithme pour la programmation linéaire. C'est alors qu'est née toute une famille d'algorithmes dits de points intérieurs. Ce type d'algorithmes permet de résoudre des problèmes d'optimisation linéaire et non-linéaire. Contrairement à la méthode du simplexe qui procède de sommets (points extrêmes, voir chapitres précédents) en sommets d'un polyèdre, les méthodes de points intérieurs s'intéressent à l'intérieur des domaines réalisables. En partant d'une région (suffisamment grande) contenant l'ensemble des solutions, il s'agit de réduire ce domaine pour cibler la solution. Ainsi, pas après pas, le domaine réalisable se restreint jusqu'à se réduire à un voisinage arbitrairement petit de la solution.

Dans ce mémoire, on s'intéresse à la résolution d'un programme linéaire en utilisant la méthode de Karmarkar. est divisé en trois chapitres :

Dans le premier, nous rappelons des notions de base de l'analyse convexe ainsi que l'optimisation sans contraintes et avec contraintes.

Le deuxième chapitre est consacré à la programmation linéaire.

Dans le dernier chapitre, étudie la méthode de karmarkar.

Chapitre 1

optimisation linéaire sans contraintes et avec constraints

1.1 Introduction

Dans Ce chapitre nous avons étudié les différents composants de mathématique et présenté les bagages nécessaires pour le traitement des problèmes considérés dans notre mémoire. Il s'agit de résultats fondamentaux d'analyse convexe et d'optimisation.

1.2 Notion de base

1.2.1 Element d'analyse convexe

Dans cette partie, nous allons rappeler les plus importants résultats d'analyse convexe nécessaire pour le développement de ce travail.

Définition 1

une partie C de \mathbb{R}^n est dite convexe si :

$$\forall x, y \in \mathbb{R}^n, \forall \lambda \in [0, 1], (1 - \lambda)y + \lambda x \in C$$

Si C est un convexe compact non vide de \mathbb{R}^n alors C admet au moins un point extrémal ce résultat puissant est dû à Minkowsky.

Définition 2

une fonction $f : \mathbb{R}^n \longrightarrow \overline{\mathbb{R}}$ est dite convexe si :

$$f[(1 - \lambda)y + \lambda x] \leq (1 - \lambda)f(y) + \lambda f(x), \forall x, y \in \mathbb{R}^n, \forall \lambda \in [0, 1]$$

f est dite strictement convexe sur \mathbb{R}^n si :

$$f[(1 - \lambda)y + \lambda x] < (1 - \lambda)f(y) + \lambda f(x), \forall x, y \in \mathbb{R}^n, x \neq y, \forall \lambda \in]0, 1[$$

f est dite concave sur \mathbb{R}^n si $-f$ est convexe sur \mathbb{R}^n . un polydère convexe borné est appelé polytope convexe.

Définition 3

une ensemble convexe de la forme :

$$P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} / A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

est dite polydère convexe.

un polydère convexe borné est appelé polytope convexe.

1.2.2 Rappels d'optimisation

Un problème d'optimisation (p) s'écrit sous la forme suivante :

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

La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée "fonction objectif", l'ensemble $\emptyset \neq C \subset \mathbb{R}^n$ est appelé "l'ensemble des solutions réalisables".

On appelle solution optimale globale de (P), tout point $\bar{x} \in C$ satisfaisant $f(\bar{x}) \leq f(x)$ $\forall x \in C$.

On dit qu'un point $\bar{x} \in C$ est "solution optimale locale" de (P), s'il existe un voisinage V de \bar{x} telle que $f(\bar{x}) \leq f(x)$, $\forall x \in C \cap V$.

1.3 Optimisation sans contraintes

Le problème d'optimisation sans contraintes s'écrit sous la forme suivante :

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

où f est une fonction de \mathbb{R}^n vers $\mathbb{R} \cup \{+\infty\}$.

1.3.1 Résultats d'existence et d'unicité

Définition 4

On dit que $f : C \rightarrow \mathbb{R}$ est coercive si $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$.

Théorème 1 (*existence*)

soit $E = \mathbb{R}^n$ et $f : E \rightarrow \mathbb{R}$ une application telle que :

1) f est continue.

2) f coercive.

alors il existe \bar{x} telle que : $f(\bar{x}) \leq f(x) \forall x \in \mathbb{R}^n$.

Remarque 1

le théorème est faux si E est de dimension infinie.

Théorème 2 (*unicité*)

soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ strictement convexe, alors il existe au plus un $\bar{x} \in \mathbb{R}^n$;
telle que :

$$f(\bar{x}) \leq f(x) \forall x \in \mathbb{R}^n.$$

Remarque 2

Ce théorème ne donne pas l'existence. Par exemple dans le cas $n = 1$ la fonction f définie par $f(x) = e^x$ n'atteint pas son minimum, car $\inf_{\mathbb{R}} f = 0$ et $f(x) \neq 0 \forall x \in \mathbb{R}$, et pourtant f est strictement convexe.

Théorème 3 (*existence et unicité*)

soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, on suppose que :

1) f continue.

2) f coercive.

3) f strictement convexe.

alors il existe un unique $\bar{x} \in \mathbb{R}^n$ telle que : $f(\bar{x}) = \inf_{\mathbb{R}^n} f$.

Remarque 3

ce théorème reste vrai si E est un espace de Hilbert.

1.3.2 Conditions d'optimalité

A) Conditions nécessaires du premier ordre :

Les conditions que nous allons donner sont des conditions différentielles qui dépendent de la dérivée de la fonction à minimiser.

Théorème 4 (*condition nécessaire d'optimalité du premier ordre*)

Soit C une partie de \mathbb{R}^n et $f : C \rightarrow \mathbb{R}$ une fonction différentiable sur C , si x^* réalise un minimum (global ou local) de f sur C alors $\nabla f(x^*) = 0$.

Définition 5

Un point x^* de C vérifiant $\nabla f(x^*) = 0$ est appelé "point critique" ou "point stationnaire".

Théorème 5

soit $f : C \rightarrow \mathbb{R}$ fonction différentiable et convexe sur C , un point x^* réalise un minimum global de f sur C si et seulement si $\nabla f(x^*) = 0$.

B) Conditions du second ordre :

Théorème 6 (*Condition nécessaire du second ordre*)

On suppose que x^* est un minimum (local) de f et que f est deux fois différentiables sur C , alors :

a) $\nabla f(x^*) = 0$.

b) $\forall x \in C \quad \langle \nabla^2 f(x^*)x, x \rangle \geq 0$.

Théorème 7 (*Condition nécessaire du seconde ordre*)

Soit f deux fois dérivable sur C vérifiant $\nabla f(x^*) = 0$ et $\exists \alpha > 0$, tel que :

$$\forall x \in C \langle \nabla^2 f(x^*)x, x \rangle \geq \alpha \|x\|^2 \dots (*)$$

alors la fonction f admet un minimum "local strict" en x^* .

1.4 Optimisation avec contraintes

La forme de optimisation avec contraintes est :

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

où C est l'ensemble des contraintes.

1.4.1 Résultats d'existence et d'unicité

Théorème 8

Supposons que f est continue, que C est un sous-ensemble fermé non vide de \mathbb{R}^n et que l'une des conditions suivantes est réalisée :

a) soit C est borné

b) soit f est coercive.

Alors le problème (P) admet au moins une solution.

Théorème 9

Sous les hypothèses du théorème 5, si f est strictement convexe et si C est convexe, alors le problème (P) admet une solution unique.

1.4.2 Condition d'optimalité

Tout comme dans le cas sans contraintes, nous allons établir des conditions d'optimalité du premier et du second ordre permettant de calculer les éventuels minima de f .

Conditions d'optimalité du premier ordre :

La condition nécessaire suivante est l'analogie de celle que nous avons dans le cas sans contraintes. Elle fait bien sûr intervenir l'ensemble des contraintes.

Théorème 10 (*Condition nécessaire du premier ordre*)

Si f est une fonction différentiable et si C est un convexe fermé, alors toute solution x^ de (P) vérifie la condition nécessaire d'optimalité du premier ordre*

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

Théorème 11 (*CNS du premier ordre dans le cas convexe*)

Supposons f convexe différentiable et C convexe fermé de \mathbb{R}^n , soit x^ un élément de C . La condition $(*)$ est nécessaire pour que x^* soit solution de (P) . Elle caractérise donc les minima de f sur C .*

On suppose maintenant que C est défini comme suit :

$$C = \{ x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p ; h_j(x) = 0, j = 1, \dots, q \}.$$

Les contraintes $g_i(x) \leq 0$ sont appelées "contraintes d'inégalités" et les contraintes $h_j(x) = 0$ "contraintes d'égalités". L'obtention de conditions nécessaires d'optimalité

nécessite que certaines conditions appelées conditions de qualification des contraintes soient vérifiées.

Ici nous nous limiterons aux trois conditions de qualification suivantes :

◆) (CQ1) C est un polyèdre convexe :

c'est le cas lorsque les fonctions g_i et h_j sont affines.

◆) (CQ2) condition de Slater :

la condition de qualification des contraintes de Slater est comme suit :

1- les fonctions g_i sont convexes et les fonctions h_j sont affines.

2- il existe \hat{x} tel que $g_i(\hat{x}) < 0$ et $h_j(\hat{x}) = 0$ pour tout i, j .

◆) (CQ3) condition de Mangasarian-Fromowitz :

la condition de qualification des contraintes de Mangasarian-Fromowitz en un point $x^* \in C$ est comme suit :

1) les q vecteurs $\nabla h_j(x^*)$ sont linéairement indépendants.

2) il existe \bar{d} tel que $\langle \nabla h_j(x^*), \bar{d} \rangle = 0$ pour tout j et $\langle \nabla g_i(x^*), \bar{d} \rangle < 0$ pour tout i .

On associe au problème d'optimisation suivant :

$$\begin{cases} \min f(x) \\ g_i(x) \leq 0, i = 1, \dots, p \\ h_j(x) = 0, j = 1, \dots, q \end{cases}$$

la fonction $L : \mathbb{R}^n \times [0, \infty[^p \times \mathbb{R}^q \longrightarrow \mathbb{R}$ définie par

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

est appelée le Lagrangien .

Le théorème suivant de Karush-Kuhn-Tucker-Lagrange donne une condition nécessaire d'optimalité :

Théorème 12

Supposons que les fonctions f, g_i, h_j sont C^1 dans un voisinage de $x^* \in C$ et que les contraintes vérifient une condition de qualification. Si f a un minimum local en x^* sur C alors il existe $\lambda \in \mathbb{R}^p$ et $\mu \in \mathbb{R}^q$ tels que :

$$\nabla_x L(x^*, \lambda, \mu) = 0, \nabla_\lambda L(x^*, \lambda, \mu) \leq 0, \nabla_\mu L(x^*, \lambda, \mu) = 0, \lambda \geq 0, \langle \lambda, \nabla_\lambda L(x^*, \lambda, \mu) \rangle = 0.$$

Les quantités λ_i et μ_j sont appelées "multiplicateurs de Karush-Kuhn-Tucker".

Lorsque le problème est convexe (f et g_i convexes et h_j affines) on a la condition nécessaire d'optimalité.

Théorème 13

Supposons que les fonctions f, g_i sont C^1 dans un voisinage de $x^ \in C$, convexes et que les fonctions h_j sont affines. S'il existe $\lambda \in \mathbb{R}^p$ et $\mu \in \mathbb{R}^q$ tels que*

$$\nabla f(x^*) + \sum_{i=1}^p \lambda_i \nabla g_i(x^*) + \sum_{j=1}^q \mu_j \nabla h_j(x^*) = 0, \lambda_i \geq 0, g_i(x^*) \leq 0, \lambda_i g_i(x^*) = 0, h_j(x^*) = 0,$$

$\forall i, j$ alors f a un minimum global en x^ sur C .*

1.5 Classification de problèmes d'optimisation

un programme mathématique (PM) est un problème d'optimisation de la forme :

$$\left\{ \begin{array}{l} \min f(x) \\ g_i(x) \leq 0, i = 1; \dots; k \\ h_j(x) = 0, j = 1, \dots m \\ x \in C \subseteq \mathbb{R}^n \end{array} \right.$$

où f, g_i, h_j sont de fonction définies de \mathbb{R}^n dans \mathbb{R} .

les problèmes d'optimisation sont classifiés selon les caractéristiques des fonctions f, g_i, h_j .

parmi les cas particulières les plus étudiées, on note :

- la programmation linéaire (f linéaire, g_i, h_j affines, C orthant positif).
- la programmation convexe (f, g_i convexes, h_j affines, C convexe).
- la programmation en nombres entiers (C discret).

Chapitre 2

Programmation linéaire

2.1 Introduction

En mathématiques, les problèmes de programmation linéaire (PL) sont des problèmes d'optimisation où la fonction objectif et les contraintes sont linéaires. Aujourd'hui, ils existent beaucoup d'applications industrielles de la programmation linéaire dans les domaines pétrolier, agro-alimentaire, l'industrie lourde, le transport et la finance.

2.2 Formes générales d'un programme linéaire

2.2.1 Forme canonique

la forme canonique s'écrit :

$$\left\{ \begin{array}{l} \min_{(x_1, \dots, x_n)} \left[F(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n = \sum_{j=1}^n c_jx_j \right] \\ \sum_{j=1}^n a_{ij}x_j \geq b_i, \forall i = 1, \dots, m \\ x_j \geq 0, \forall j = 1, \dots, n \end{array} \right.$$

et l'écriture matricielle est de la forme canonique est :

$$\begin{cases} \min c^t x \\ Ax \geq b \\ x \geq 0 \end{cases}$$

2.2.2 Forme standard

la forme standard s'écrit :

$$\begin{cases} \min_{(x_1, \dots, x_n)} \left[F(x_1, \dots, x_n) = c_1 x_1 + \dots + c_n x_n = \sum_{j=1}^n c_j x_j \right] \\ \sum_{j=1}^n a_{ij} x_j = b_i, \forall i = 1, \dots, m \\ x_j \geq 0, \forall j = 1, \dots, n \end{cases}$$

et l'écriture matricielle est de la forme standard est :

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

Proposition 1 *Tout PL sous forme standard s'écrit de façon équivalente en un PL sous forme canonique et inversement.*

2.3 Equivalence de deux programmes mathématiques

Deux programmes mathématiques (en particulier deux programmes linéaires) sont dits "équivalents" si à toute solution réalisable de l'un on sait faire correspondre une solution réalisable de l'autre, de telle sorte que les fonctions objectives soient égales pour cette paire de solution.

2.4 dualité

2.4.1 La dualité en programmation linéaire

A chaque programme linéaire est associé un autre programme linéaire appelé dual.

Définition 6

Soit le programme linéaire écrit sous forme standard :

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

$c, x \in \mathbb{R}^n, A \in M_{m,n}(\mathbb{R}), b \in \mathbb{R}^m$.

On appellè dual du (primal) le programme linéaire :

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

- 1) Le dual du dual est le primal.
- 2) À partir de la relation primale-duale définie pour la forme standard, il est possible de retrouver le programme dual de tous les programmes linéaires non formulés sous cette forme.
- 3) les variables du dual sont en bijection avec les contraintes du primal.
- 4) les contraintes du dual sont en bijection avec les variables du primal.

Soit le problème linéaire suivant, considéré comme étant le primal.

$$\left\{ \begin{array}{l} \min(x_1 + x_2) \\ x_1 - x_2 = 0 \\ x_1 + x_2 + x_3 = 1 \\ x_i \geq 0, i = 1, 2, 3 \end{array} \right.$$

Le dual de ce programme s'écrira :

$$\left\{ \begin{array}{l} \max y_2 \\ y_1 + y_2 \leq 1 \\ -y_1 + y_2 \leq 1 \\ y_2 \leq 0 \\ y \in \mathbb{R}^2 \end{array} \right.$$

Exemple 1 soit le programme suivant :

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

la dual de ce programme est :

$$\left\{ \begin{array}{l} \max b^t y \\ A^t y \leq c \\ y \geq 0 \end{array} \right.$$

Proposition 2 (*dualité faible*)

Si x est primal réalisable et y est dual réalisable alors $b^t y \leq c^t x$.

Proposition 3 (*dualité forte*)

Si x est optimal pour le programme primal, alors le programme dual a une solution optimal y telle que $b^t y = c^t x$.

Proposition 4 *Soit x primal réalisable et y dual réalisable avec $s = c - A^t y$, alors on a :*

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

Remarque 4

Si l'objectif d'un problème (dual ou primal) n'est pas borné, l'autre problème n'admet pas de solution réalisable.

2.4.2 Application de la dualité

terminons ce paragraphe en signalant l'importance de la dualité aussi bien sur le plan pratique que théorique.

en effet, en plus de l'éclairage qu'elle apporte sur certains points d'ordre théorique permettant de mieux caractériser les propriétés du problème donné, les résultats précédents montrent qu'en résolvant un programme linéaire primal, on résout en même temps son dual.

Chapitre 3

Méthode de karemarkare

3.1 Introduction

En 1984, Karmarkar a proposé un algorithme prometteur pour la programmation linéaire, il possède des propriétés théoriques attractives et un bon comportement numérique.

L'algorithme est conçu pour résoudre un programme linéaire de la forme :

$$\left\{ \begin{array}{l} \min c^t x = z^* \\ Ax = 0 \quad \dots\dots\dots(p) \\ x \in S_n \end{array} \right.$$

3.2 Préliminaires

Minimisation d'une fonction linéaire sur un ellipsoïde :

Considérons le problème suivant :

$$\left\{ \begin{array}{l} \min c^t x \quad \dots\dots\dots(O) \\ x \in E \end{array} \right.$$

où $x, c \in \mathbb{R}^n$ et E un ellipsoïde défini par son centre \hat{a} , par r et la matrice symétrique

définie positive A .

Algébriquement :

$$E = \{ x \in \mathbb{R}^n : (x - \hat{a})^t A (x - \hat{a}) \leq r^2 \}$$

Lemme 1

La solution du problème (O) est donnée par :

$$x^* = \hat{a} - \frac{r}{\sqrt{\langle c, A^{-1}c \rangle}} * A^{-1}c$$

Cas particulier du lemme 1 :

Si l'ellipsoïde E est réduit à une sphère ($A = I$) alors la solution de (O) est :

$$x^* = \hat{a} - \frac{c}{\|c\|} r.$$

Autrement dit, il suffit de se déplacer à partir du centre \hat{a} dans la direction opposée du vecteur coût (i.e $-c$) d'une distance égale au rayon (r) de la sphère.

3.3 L'idée générale de l'algorithme

Soit A un élément de $\mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$ et posons par définition

$$\Omega = \{x \in \mathbb{R}^n : Ax = b\}$$

$$P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} = \Omega \cap \mathbb{R}_+^n$$

Considérons alors le programme linéaire général :

$$\begin{cases} \min c^t x \\ x \in P \end{cases} \dots\dots\dots(p.l.g)$$

Si l'on remplace \mathbb{R}_+^n par une sphère (ou un ellipsoïde) $E_1 \subset \mathbb{R}_+^n$ on obtient le problème :

$$\begin{cases} \min c^t x \\ x \in \Omega \cap E_1 \end{cases} \dots\dots\dots(1)$$

Or, l'intersection d'une sphère et d'un espace affine est une sphère de dimension plus petite dans cet espace affine, donc $(\Omega \cap E_1 = E_2)$.

Le problème devient :

$$\begin{cases} \min c'^t x \\ x \in E_2 \end{cases} \dots\dots\dots(2)$$

c' désigne la projection orthogonale de c sur Ω .

D'après ce qui précède, le problème (2) est trivial : Il suffit de se déplacer à partir du centre de E_2 dans la direction $(-c')$ d'une distance égale au rayon de E_2 .

Finalement, tout revient à remplacer P par un sphère (ou une ellipsoïde) E de centre \hat{a} ($\hat{a} \in P : \hat{a}_i > 0, i = 1, \dots, n$) et minimiser $c^t x$ sur E au lieu de P . La question est donc de savoir quelle sera la qualité de la solution obtenue comparée à la solution réelle du problème.

Pour donner une réponse à la question, on construit un deuxième ellipsoïde E' contenant P de même centre que E et de rayon égal à ν fois le rayon de E ($E \subset P \subset E'$), ν étant un réel strictement plus grand que 1.

Notons par $f_E = f(\hat{a})$, $f_P = f(x_P)$ et $f_{E'} = f(x_{E'})$ les minima de $f(x) = c^t x$ sur E , P , E' respectivement.

On a alors le résultat suivant :

Lemme 2

$$0 \leq \frac{f_{E'} - f_P}{f(\hat{a}) - f_P} \leq 1 - \frac{1}{\nu}.$$

En partant donc de \hat{a} , on arrive à un point a^1 minimisant f sur E ($f(a^1) = f_P$) et f_P est approchée par un facteur de $(1 - \frac{1}{\nu})$. On recommence l'opération en choisissant a^1 comme centre d'un sphère E_1 contenu dans P et d'un sphère $E'_1 = \nu_1 E_1$ contenant P . Ainsi :

$$0 \leq f_E - f_P \leq (1 - \frac{1}{\nu_1})(1 - \frac{1}{\nu})[f(\hat{a}) - f_P].$$

On peut encore réitérer avec un point a^2 centre d'un ellipsoïde E_2 et ainsi de suite....Le but étant de borner $f_E - f_P$ par une valeur de plus en plus petite jusqu'à atteindre une précision voulue. Bien entendu, la vitesse de convergence dépend de ν_i : Plus ce dernier est petit plus la convergence est rapide.

3.4 Programme linéaire traité par Karmarkar

La méthode de Karmarkar est destinée à résoudre les problèmes de la forme :

$$\left\{ \begin{array}{l} \min c^t x = z^* = 0 \\ Ax = 0 \quad \dots\dots\dots(p.l.r) \\ x \in S_n = \{x \in \mathbb{R}^n / e_n^t x = 1, x \geq 0\} \end{array} \right.$$

où $c \in \mathbb{R}^n$ (vecteur coût), A est une matrice $m \times n$ réelle de plein rang ($rgA = m < n$) et $e_n^t = (1, 1, \dots, 1)^t \in \mathbb{R}^n$.

Ayant comme solution strictement réalisable $\hat{x} = \frac{e_n}{n}$, le centre du simplexe S_n .

Remarque 5

1) Si le système des contraintes n'est pas homogène ($Ax = b$ et $b \neq 0$) ou si la valeur optimale est connue et non nulle, l'égalité $e_n^t x = 1$ permet de se ramener facilement à la forme (p.l.r). En effet, il suffit d'écrire : $Ax = b$ sous la forme :

$$(Ax = be_n^t x) \implies [(A - be_n^t)x = 0]$$

$$\text{avec } \tilde{A} = A - be_n^t$$

ainsi, l'opérateur précédent devient : $\tilde{A}x = 0$.

De la même manière si $c^t x = z^*$ multiplions le 2^{ème} membre de l'équation précédente par : $e_n^t x = 1$ pour obtenir :

$$(c^t x - z^* e_n^t x = 0) \implies [(c^t - z^* e_n^t)x = 0]$$

posons :

$$(\tilde{c}^t = c^t - z^* e_n^t) \implies (\tilde{c}^t x = 0).$$

2) L'ignorance de la valeur optimale ne constitue pas une difficulté car il existe plusieurs variantes qui approximent cette valeur.

3) Le problème (p.l.r) peut être vu comme suit : Trouver $x \geq 0$ tel que :

$$\left\{ \begin{array}{l} c^t x = 0 \\ Ax = 0 \\ e_n^t x = 1 \\ x \geq 0 \end{array} \right.$$

3.5 Transformation projective de Karmarkar

D'après les opinions de plusieurs auteurs, la transformation projective joue un grand rôle pour comprendre les méthodes de points intérieurs.

Pour cela, Karmarkar utilise à chaque itération la transformation projective T_k :

$$T_k : S_n \longrightarrow S_n \\ x \longmapsto y = T_k(x)$$

Où

$$T_k(x) = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x} \quad \text{et} \quad x = T_k^{-1}(y) \quad \text{avec} \quad T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}.$$

Dans ce qui suit, on présente l'algorithme de Karmarkar pour résoudre un problème de la forme (p.l.r) :

Part de la solution initiale \hat{x} , l'algorithme construit une suite de points intérieurs qui converge vers une solution optimale du problème (p.l.r).

Pour ramener l'objectif ($c^t x$) à zéro, on le projete par la projection P_k et on le minimise localement sur une sphère inscrite dans la région réalisable du problème (p.l.r).

À chaque itération (k), l'itéré ($x^k > 0$) est renvoyé au centre du simplexe S_n par la transformation T_k et ainsi de suite, jusqu'à ce que le test d'optimalité ($c^t x \leq \varepsilon$) soit vérifié.

3.6 Description de l'algorithme de base

Partant de la solution initiale $x^0 = a$; l'algorithme construit une suite de points intérieurs qui converge vers une solution optimale du problème en un temps polynômial.

Dans le but de ramener facilement l'objectif à zéro, on le minimise localement sur une sphère inscrite dans la région réalisable.

Donc, à chaque itération k ; l'itéré $x^k > 0$ est ramené au centre de S_n par la transformation projective T_k définie par :

$$T_k : x \in S_n \longrightarrow T_k(x) = y \in S_n$$

avec :

$$T_k(x) = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x} = y, \quad T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}.$$

où $D_k = \text{diag}\{x^k\}$.

La transformation T_k applique le simplexe S_n dans lui même, en même temps l'itéré $x^k > 0$ est envoyé au centre de S_n , le transformé du programme linéaire (PLS) est le programme fractionnaire suivant :

$$\left\{ \begin{array}{l} \min \frac{c^t D_k y}{e_n^t D_k y} \\ \frac{A D_k y}{e_n^t D_k y} = 0 \\ e_n^t y = 1, \\ y \geq 0 \end{array} \right. \dots\dots\dots(3)$$

Les hypothèses de départ permettent d'obtenir le programme linéaire équivalent suivant :

$$\left\{ \begin{array}{l} \min c^t D_k y \\ A D_k y = 0 \\ e_n^t y = 1, \\ y \geq 0 \end{array} \right. \dots\dots\dots(4)$$

qui est bien de la forme (PLS).

On a besoin du lemme suivant pour simplifier la résolution du problème (4) ci-dessus :

Si pour un programme linéaire donné on connaît une solution réalisable y^0 tel que :
 $(y_i^0 > 0, i = 1 : n + 1)$, alors l'ellipsoïde : $E = \left\{ y \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} \frac{(y_i - y_i^0)}{(y_i^0)^2} \leq \beta^2, 0 < \beta < 1 \right\}$
 est dans l'intérieur de l'orthant positif de \mathbb{R}^{n+1} .

Théorème 14

La solution optimale du problème (p) est donnée explicitement par :

$$y^k = a - \alpha r d^k$$

$$\text{où } d^k = \frac{p^k}{\|p^k\|} \text{ et } p^k = p_{B_k}(D_k c), B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$$

3.6.1 Algorithme de base

Début algorithme

Initialisation : $\varepsilon > 0$ est une précision donnée, $x^0 = a = \frac{1}{n} e_n, k = 0$

Pas 1 :

Tant que : $c^t x^k > \varepsilon$ faire :

- construire $D_k = \text{diag}\{x^k\}, A_k = A D_k, B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$

- calculer $p^k = (I - B_k^t (B_k B_k^t)^{-1} B_k) D_k c, d^k = \frac{p^k}{\|p^k\|}$

- calculer $y^k = a - \alpha r d^k, r = \frac{1}{\sqrt{n(n-1)}}, 0 < \alpha < 1$

Pas 2 :

- prendre $x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k}{e_n^t D_k y^k}, k = k + 1$ et retourner au pas 1.

Fin Tant que

Fin algorithme.

Remarque 6

Le calcul de la projection P_k constitue l'opération la plus coûteuse dans l'algorithme. L'efficacité pratique de l'algorithme dépend, en grande partie, de la manière du calcul de P_k . De même la vitesse de convergence de l'algorithme dépend du pas de déplacement.

3.6.2 Dérivation et analyse de l'algorithme

Le but de ce paragraphe est de montrer en quelque sorte comment est obtenu l'algorithme précédent.

On a vu que la transformation T_k applique le simplexe S_n dans lui même, en même temps l'itéré $x^k > 0$ (dont les composantes forment la matrice diagonale D_k) est envoyé au centre de S_n . Cependant le transformé du programme linéaire (p.l.r) est le programme suivant :

$$\left\{ \begin{array}{l} \min \frac{c^t D_k y}{e_n^t D_k y} \\ \frac{AD_k y}{e_n^t D_k y} = 0 \quad \dots\dots\dots(p.n.l) \\ e_n^t y = 1, y \geq 0 \end{array} \right.$$

Mais on a pour tout $y \in S_n$:

$$e_n^t D_k y = \sum_{i=1}^n x_i^k y_i \geq \min\{x_i^k; i = 1, \dots, n\} > 0.$$

Les égalités

$$\frac{c^t D_k y}{e_n^t D_k y} = 0 \quad \text{et} \quad \frac{AD_k y}{e_n^t D_k y} = 0$$

sont donc satisfaites si et seulement si :

$$c^t D_k y = 0 \quad \text{et} \quad AD_k y = 0$$

(p.n.l) est alors équivalent au programme linéaire :

$$\left\{ \begin{array}{l} \min c^t D_k y \\ AD_k y = 0 \quad \dots\dots\dots(p.l) \\ e_n^t y = 1; y \geq 0 \end{array} \right.$$

qui est de la forme (p.l.r) et vérifie les conditions de Karmarkar.

Une remarque fondamentale :

Si on ajoute au problème (p.1) la contrainte $\{y \in \mathbb{R}^n : \|\hat{a} - y\| \leq \alpha r\}$ où $0 < \alpha < 1$ et $r = \frac{1}{\sqrt{n(n-1)}}$ i.e la sphère $B(\hat{a}, \alpha r)$ de centre \hat{a} et de rayon αr , la contrainte de positivité ($y \geq 0$) devient redondante.

Ce résultat est une conséquence du lemme général suivant :

Lemme 3

Si pour un programme linéaire donné, on connaît une solution réalisable y telle que ($y_i > 0, i = 1, \dots, n$) alors l'ellipsoïde :

$$E = \{x \in \mathbb{R}^n : \sum_{i=1}^n \frac{(x_i - y_i)^2}{y_i^2} \leq \beta^2, 0 < \beta < 1\}$$
 est à l'intérieur de l'orthant positif de \mathbb{R}^n .

Pour retrouver le résultat précédent il suffit de prendre $y = \hat{a}$ et $\beta = \alpha r$.

Finalement le problème (p.1) devient :

$$\left\{ \begin{array}{l} \min c^t D_k y \\ AD_k y = 0 \\ e_n^t y = 1 \\ \|y - \hat{a}\| \leq \alpha r \end{array} \right. \dots\dots\dots (p_{aux})$$

On vient de remplacer l'orthant positif par la sphère $B(\hat{a}, \alpha r)$ qui est beaucoup plus simple à manier.

Le point y^k (du pas 3 de l'algorithme) est une solution optimale de (p_{aux}).

Posons $x = y - \hat{a}$, alors $B_k x = 0$ où B_k est la matrice des contraintes de (p_{aux}) définie au pas 0 de l'algorithme. (p_{aux}) est alors équivalent (à une translation près) au problème suivant :

$$\left\{ \begin{array}{l} \min c^t D_k x \\ B_k x = 0 \\ \|x\|^2 = \sum_{i=1}^n x_i^2 \leq (\alpha r)^2 \end{array} \right. \dots\dots\dots (p^*)$$

Or x^* est une solution optimale de (p^*) si et seulement si $\exists \lambda \in \mathbb{R}^{m+1}$ et un scalaire $\mu \geq 0$ tel que : $D_k c + B_k^t \lambda + \mu x^* = 0 \dots \dots \dots (i)$.

En multipliant les deux membres de (i) par B_k , on trouve :

$$B_k D_k c + B_k B_k^t \lambda + \mu B_k x^* = B_k D_k c + B_k B_k^t \lambda = 0$$

(puisque $B_k x = 0$) $\implies \lambda = -(B_k B_k^t)^{-1} (B_k D_k c)$ d'où en substituant dans (i)

$$x^* = \left(-\frac{1}{\mu}\right) [I - B_k^t (B_k B_k^t)^{-1} B_k] D_k c = \left(-\frac{1}{\mu}\right) P_k \text{ mais } x^* \text{ vérifie :}$$

$$\|x^*\| = \frac{1}{\mu} \|P_k\| = \alpha r \implies \frac{1}{\mu} = \frac{\alpha r}{\|P_k\|} \implies x^* = -\alpha r \frac{P_k}{\|P_k\|} = -\alpha r d_k \text{ soit finalement :}$$

$$y^k = y^* = \hat{a} + x^* = \hat{a} - \alpha r d_k.$$

3.7 Convergence de l'algorithme

La fonction objective du problème transformé se réduit d'un montant de $\left(1 - \frac{\alpha}{n-1}\right)$ conformément au théorème suivant :

.le point y^k vérifie : $\frac{c^t D_k y^k}{c^t D_k a} \leq 1 - \frac{\alpha}{n-1}$.

Pour établir la convergence de cet algorithme, Karmarkar introduit à l'objectif la fonction potentiel :

$$f(x) = \sum_{i=1}^n \ln\left(\frac{c^t x}{x_i}\right) \text{ définie sur } \{x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1\}.$$

Le lemme suivant montre que la réduction de $f(x)$ conduit directement à celle de $c^t x$.

Soit x^k le k^{ieme} itéré de l'algorithme, alors :

$$\frac{c^t x^k}{c^t x^0} \leq \left(\exp [f(x^k) - f(x^0)]\right)^{\frac{1}{n}}.$$

Donc, si la suite $(f(x^k))$ tend vers $-\infty$ alors la suite $(c^t x^k)$ tend vers zéro.

Dans le théorème suivant, Karmarkar montre que la convergence de son algorithme est réalisée en $O(nq + n \ln n)$ itérations pour $0 < \alpha \leq \frac{1}{4}$.

si. $0 < \alpha \leq \frac{1}{4}$, alors en partant de $x^0 = \frac{1}{n} e_n$, l'algorithme trouve après $O(nq + n \ln n)$

itérations un point réalisable x tel que :

$$1/ c^t x = 0$$

ou

$$2/ \frac{c^t x}{c^t x^0} \leq \varepsilon = 2^{-q} \text{ où } q \text{ est une précision fixée.}$$

3.8 Généralisation de l'algorithme de Karmarkar

Soit le programme linéaire sous la forme standard :

$$\left\{ \begin{array}{l} \min c^t x = z^* \\ Ax = b \quad \dots\dots\dots(p) \\ x \geq 0 \end{array} \right.$$

A est une matrice de type (m,n) , $c \in \mathbb{R}^n$ est le vecteur coût et $b \in \mathbb{R}^m$.

On définit le simplexe S_{n+1} de dimension n contenu dans \mathbb{R}^{n+1} par :

$$S_{n+1} = \{x \in \mathbb{R}^{n+1} : x \geq 0, e_{n+1}^t x = 1\}$$

On suppose que :

- (H1) La matrice A est de plein rang ($\text{rg}A = m \leq n$).
- (H2) On dispose d'un point x^0 strictement réalisable ($Ax^0 = b, x^0 > 0$).
- (H3) La valeur optimale z^* de l'objectif est connue au départ.

L'hypothèse 1 est classique. L'hypothèse 2 n'est pas du tout restrictive, car on peut toujours obtenir une solution strictement réalisable ou prouver que le problème n'est pas réalisable moyennant une méthode simple de variable artificielle . Pour l'hypothèse 3, Si la valeur optimale est non nulle l'égalité $e_{n+1}^t x = 1$ permet de se ramener à un objectif nul. En effet, soit x^* une solution optimale du problème et z^* la valeur optimale de l'objectif. Alors :

$$c^t x^* = z^* = z^* e_{n+1}^t x^* \implies (c - z^* e_{n+1})^t x^* = \hat{c}^t x = 0.$$

Si z^* n'est pas connue, il existe des techniques d'approximation fiables permettant de remplacer z^* par des bornes inférieures ou supérieures convenables.

Pour le système de contraintes : $Ax = b$; $b \neq 0$; on se ramène facilement à un système homogène, il suffit d'écrire :

$$Ax = be_{n+1}^t x \implies (A - be_{n+1}^t)x = 0.$$

La transformation projective notée T_k est une fonction

$$T_k : \mathbb{R}_+^n \implies S_{n+1} \text{ définie par : } T_k(x) = y$$

avec :

$$\left\{ \begin{array}{l} y_i = \frac{\frac{x_i}{x_i^k}}{1 + \sum_{i=1}^n \frac{x_i}{x_i^k}} \text{ avec } i = 1, \dots, n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{array} \right. \dots\dots\dots(1^*)$$

on a :

$$y_i = \frac{x_i}{x_i^k} y_{n+1}, i = 1, \dots, n$$

ou encore $y[n] = (D_k^{-1}x)y_{n+1}$ où $y[n]$ désigne les n premières composantes de y et on

$$a : x = T_k^{-1}(y) = \frac{D_k y[n]}{y_{n+1}}, D_k = \text{diag}(x^k).$$

On applique la transformation T_k au programme (1*), on obtient :

$$\left\{ \begin{array}{l} \min \frac{c^t D_k y[n]}{y_{n+1}} \\ \frac{AD_k y[n]}{y_{n+1}} = 0 \\ \sum_{i=1}^{n+1} y_i = 1 \\ y[n] \geq 0, y_{n+1} > 0 \end{array} \right. \dots\dots\dots(2^*)$$

ou encore :

$$\left\{ \begin{array}{l} \min \hat{c}^t y \\ \hat{A}y = 0 \\ e_{n+1}^t y = 1 \\ y \geq 0 \end{array} \right. \dots\dots\dots(3^*)$$

Où

$$\left\{ \begin{array}{l} \hat{c}_i = c_i x_i^k, i = 1, \dots, n \\ \hat{c}_{n+1} = -z^* \end{array} \right. , y = \begin{bmatrix} y[n] \\ y_{n+1} \end{bmatrix} \text{ et } \hat{A} = [AD_k - b].$$

Notons que toute solution réalisable de (1*) est transformée par T_k en une solution réalisable de (3*) et réciproquement, toute solution réalisable y de (3*) avec $y_{n+1} > 0$ est transformée par T_k^{-1} en une solution réalisable x de (1*).

3.9 Extension de la méthode de Karmarkar

Considérons le problème d'optimisation non linéaire suivant :

$$\begin{cases} \min f(x) \\ Ax = b \quad \dots\dots\dots(1) \\ x \geq 0 \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, est une fonction non linéaire, convexe et différentiable, A une matrice de type (m,n) , avec les hypothèses suivantes :

- (H1) La matrice A est de plein rang ($\text{rg}A = m \leq n$) :
- (H2) On dispose d'un point x^0 strictement réalisable ($Ax^0 = b$; $x^0 > 0$).
- (H3) La valeur optimale z^* de l'objectif est connue au départ.

3.9.1 Formulation du problème

On commence par ramener le problème à la forme simplifiée suivante :

$$\begin{cases} \min g(y) = 0 \\ By = 0 \quad \dots\dots\dots(2) \\ y \in S_{n+1} \end{cases}$$

où $g : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$, est une fonction non linéaire, convexe et différentiable.

$S_{n+1} = \{y \in \mathbb{R}^{n+1} : e_{n+1}^t y = 1, y \geq 0\}$ est le simplexe de dimension n et de centre a tel que : $a_i = \frac{1}{1+n}, \forall i \in \{1, \dots, n+1\}$. $e_{n+1} \in \mathbb{R}^{n+1}$ où $e_i = 1; \forall i \in \{1, \dots, n+1\}$.

En effet, en appliquant la transformation projective T_k définie par :

$$T_k : \mathbb{R}^n \rightarrow S_{n+1} \text{ tel que : } T_k(x) = y$$

avec

$$\begin{cases} y_i = \frac{\frac{x_i}{x_i^k}}{1 + \sum_{i=1}^n \frac{x_i}{x_i^k}} & \text{avec } i = 1, \dots, n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{cases}$$

$$x = T_k^{-1}(y) = \frac{D_k y[n]}{y_{n+1}} \text{ où } y[n] = D_k^{-1} x, y_{n+1} = (y_i)_{i=1}^n, D_k = \text{diag}(x^k).$$

Le problème :

$$\begin{cases} \min f(x) = z^* \\ AX = 0 \\ x \geq 0 \end{cases}$$

se transforme comme suit

$$\begin{cases} \min [f(T_k^{-1}(y)) - z^*] = f\left(\frac{D_k y[n]}{y_{n+1}}\right) - z^* \\ \frac{AD_k y[n]}{y_{n+1}} = b \\ \sum_{i=1}^{n+1} y_i = 1 \\ y[n] \geq 0, y_{n+1} > 0 \end{cases} \dots\dots\dots(3)$$

Pour avoir la convexité de la fonction objectif, il convient d'écrire (3) sous la forme équivalente :

$$\begin{cases} \min g(y) = y_{n+1} [f(T_k^{-1}(y)) - z^*] \\ A_k y = 0 \\ y \in S_{n+1} \end{cases} \dots\dots\dots(4)$$

$$A_k = [AD_k - b], y = \begin{bmatrix} y[n] \\ y_{n+1} \end{bmatrix}$$

Notons que la valeur optimale de g est 0 et que le centre du simplexe est réalisable pour (4).

Notons aussi que la fonction g est convexe sur l'ensemble : $Y = \{y \in \mathbb{R}^{n+1} : A_k y = 0, y \in S_{n+1}\}$ conformément au lemme suivant :

La fonction f étant convexe sur l'ensemble $X = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, il en est de même pour g sur l'ensemble $Y = \{y \in \mathbb{R}^{n+1}; A_k y = 0, y \in S_{n+1}\}$

3.9.2 Linéarisation

En linéarisant la fonction g au voisinage du point a (centre du simplexe) et en introduisant une boule de centre a considérée comme voisinage de a :

$$g(y) = g(a) + \langle \nabla g(a), y - a \rangle \text{ pour } y \in \{y \in \mathbb{R}^{n+1} : \|y - a\| \leq \alpha\}.$$

On obtient le sous problème suivant :

$$\left\{ \begin{array}{l} \min \nabla g(a)^t y \\ A_k y = 0 \\ e_{n+1}^t y = 1, y \geq 0 \\ \|y - a\|^2 \leq \alpha^2 \end{array} \right. \dots\dots\dots(5)$$

On sait que pour $\alpha < 1$ la contrainte $y \geq 0$ est redondante et donc :

$$\left\{ \begin{array}{l} \min \nabla g(a)^t y \\ A_k y = 0 \\ e_{n+1}^t y = 1 \\ \|y - a\|^2 \leq \alpha^2 \end{array} \right. \dots\dots\dots(6)$$

La solution optimale du problème (6) est donnée explicitement par : $y^k = a - \alpha d^k$ où $d^k = \frac{p^k}{\|p^k\|}$ et $p^k = p_{B_k}(\text{rg}(a))$, $B_k = \begin{bmatrix} A_k \\ e_{n+1}^t \end{bmatrix}$

3.9.3 Description de l'algorithme

Partant d'une solution initiale x^0 strictement réalisable, à chaque itération k , on applique la transformation projective T_k pour envoyé le point courant x^k au centre du

simplexe, on calcule le point y^k solution optimale du problème linéarisé. Enfin, on revient à la variable initiale en appliquant T_k^{-1} et on s'arrête si le test d'optimalité est vérifié.

Il suffit alors de contrôler la réduction de l'objectif du problème initial pour établir la convergence de l'algorithme correspondant que nous décrivons comme suit :

Algorithme

Début algorithme

Initialisation : $\varepsilon > 0$ est une précision donnée, x^0 est un point strictement réalisable.

Pas 1 :

Tant que : $f(x^k) - z^* > \varepsilon$ faire :

- construire $D_k = \text{diag}\{x^k\}$, $A_k = [AD_k - b]$, $B_k = \begin{bmatrix} A_k \\ e_{n+1}^t \end{bmatrix}$
- calculer $p^k = (I - B_k^t (B_k B_k^t)^{-1} B_k) \nabla g(a)$, $d^k = \frac{p^k}{\|p^k\|}$
- calculer $y^k = a - \alpha d^k$

Pas 2 :

- prendre $x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k [n]}{y_{n+1}^k}$, $k = k + 1$ et retourner au pas 1.

Fin Tant que

Fin algorithme.

3.9.4 Etude de la convergence

Pour établir la convergence de notre algorithme, on introduit la fonction potentiel associée au problème (1) définie par :

$$P(x) = (n+1) \ln(f(x) - z^*) - \sum_{i=1}^n \ln(x_i).$$

Notre but est de réduire la fonction potentiel $P(x)$ pour avoir une réduction dans la fonction $(f(x) - z^*)$.

$$\begin{aligned} \text{on a : } P(x^k) - P(x^0) &= [(n+1) \ln(f(x^k) - z^*) - \sum_{i=1}^n \ln(x_i^k)] - [(n+1) \ln(f(x^0) - z^*) - \sum_{i=1}^n \ln(x_i^0)] \\ &= (n+1) \ln \left[\frac{f(x^k) - z^*}{f(x^0) - z^*} \right] - \left[\sum_{i=1}^n \ln(x_i^k) - \sum_{i=1}^n \ln(x_i^0) \right] \\ \frac{P(x^k) - P(x^0)}{n+1} &= \ln \left[\frac{f(x^k) - z^*}{f(x^0) - z^*} \right] - \left[\frac{\sum_{i=1}^n \ln(x_i^k) - \sum_{i=1}^n \ln(x_i^0)}{n+1} \right] \end{aligned}$$

$$\frac{f(x^k) - z^*}{f(x^0) - z^*} = v(x^k) \exp\left(\frac{P(x^k) - P(x^0)}{n+1}\right)$$

avec $v(x^k) = \exp\left(\frac{\sum_{i=1}^n \ln(x_i^k) - \sum_{i=1}^n \ln(x_i^0)}{n+1}\right)$

le lemme suivant montre qu'on a une réduction dans la fonction g .

Lemme 4

On a : $g(y^k) < g(a)$, $g(y^k) \leq \left(1 - \frac{\alpha}{n+1}\right) g(a)$

où y^k est la solution optimale du problème (6).

A chaque itération de l'algorithme, la fonction potentiel se réduit d'une valeur constante telle que : $P(x^{k+1}) \leq P(x^k) - \delta$, où $\delta = \alpha - \frac{\alpha^2}{2(1-\alpha)^2}$

L'algorithme correspondant à cette variante est de structure simple et nous avons de bonnes raisons pour prétendre un comportement numérique meilleur.

3.9.5 Fonction potentielle et convergence

Pour établir la convergence de l'algorithme, il faut montrer que :

$$\frac{c^t x^{k+1}}{c^t x^k} < q_0$$

où $0 < q_0 < 1$ et indépendant de k .

Or, il est difficile de trouver directement q_0 .

Pour surmonter cette difficulté, Karmarkar associe à la fonction linéaire $c^t x$ la fonction potentielle logarithmique suivante :

$$f(x) = \sum_{i=1}^n \log\left[\frac{c^t x}{x_i}\right]$$

définie sur l'ensemble de la stricte réalisabilité

$$F^0 = \{x \in \mathbb{R}^n, x > 0, Ax = 0, e_n^t x = 1\}.$$

Le lemme suivant montre que la minimisation (réduction) de $f(x)$ conduit droit à celle de $c^t x$.

Lemme 5

Soit x^k le k^{eme} itéré de l'algorithme, alors :

$$\frac{c^t x^k}{c^t x^0} \leq (\exp[f(x^k) - f(x^0)]^{\frac{1}{n}} \text{ où } x^0 = \frac{e_n}{n}.$$

Conséquence immédiate :

Si la suite $f(x^k)$ tend vers $-\infty$ alors la suite $c^t x^k$ tend vers zéro autrement dit, pour diminuer $c^t x$, il suffit de diminuer suffisamment $f(x)$.

3.10 Calcul d'une solution initiale strictement réalisable

Un point strictement réalisable de (p) est une solution du problème

$$(SR) : \left\{ \begin{array}{l} \text{trouver } x \\ Ax = b, x > 0 \end{array} \right\}.$$

En utilisant la technique de la variable artificielle, le problème (SR) est équivalent au problème (p) suivant :

$$\left\{ \begin{array}{l} \min \lambda \\ Ax + \lambda(b - Aa) = b \\ x > 0, \lambda \geq 0 \end{array} \right.$$

tel que $a > 0$ est choisi arbitrairement dans l'orthant positif et λ une variable artificielle.

Posons $y = (x, \lambda)$, (s) est équivalent au programme linéaire :

$$\begin{cases} \min c^t y = z^* = 0 \\ A'y = b \\ y \geq 0 \end{cases}$$

Où $c = (0, \dots, 0, 1)^t \in \mathbb{R}^{n+1}$, $A' = (A, b - Aa) \in \mathbb{R}^{m \times (n+1)}$.

(s) vérifie les hypothèses de Karmarkar :

1) $z^* = 0$.

2) $y = (a, 1)$ solution strictement réalisable de (p).

3) la matrice A' est de plein rang, $rg(A') = m < n + 1$ la résolution de (SR) se ramène à celle de (s).

Plus précisément, Karmarkar démontre le théorème suivant :

Théorème 15 $\exists \varepsilon_0 > 0$ tel que : Les deux propositions suivantes sont équivalentes :

1) x solution strictement réalisable de (SR).

2) (s) admet une solution optimale (x, λ) telle que $\lambda \leq \varepsilon_0$.

La solution optimale de (SR) représente une solution réalisable de (p), ce qui règle le problème d'initialisation posé, cette étape s'appelle la phase1 de Karmarkar.

conclusion

Durant l'analyse de méthode karmarkar, nous avons remarque que cet dernier basèe sur la transformation de polidère à simplex pour obtenir un optimisation de ce probleme, mais malgré la reussite de cette méthode, il reste le probleme de temps de convergence de l'algorithme.

Bibliographie

- [1] J-B. Hiriart-Urruty, Optimisation et analyse convexe. PUF, (1998).
- [2] M. AZI, Cours optimisation, Centre universitaire de mila, 2013-2014.
- [3] J.P.BRANS, Optimisation mathématique, Bruxelles, ULB, 1996.
- [4] S. EL BERNOUSSI, Programmation lineaire Methode du simplexe, 2010.
- [5] A. Keraghel, "Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar". Dissertation thesis (1989), Université Joseph Fourier, Grenoble, France, (1989).
- [6] A. Keraghel, D. Benterki, "sur les performances de l'algorithme de Karmarkar pour la programmation linéaire", Revue Roumaine des sciences techniques-Mécanique appliquées, Tome 46, n°.1 (2001).
- [7] Grégoire Allaire, Analyse numérique et optimisation. France, 2008.
- [8] J-C. Culioli, Introduction à l'optimisation, France, 2012.
- [9] [Fr.wikipedia.org/wiki/Algorithme_de_Karmarkar](http://fr.wikipedia.org/wiki/Algorithme_de_Karmarkar).