

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS-SETIF

## **THESE**

Présentée à la Faculté des Sciences  
Département de Mathématiques  
Pour L'Obtention du Diplôme de

## **DOCTORAT D'ETAT**

**Option : Mathématiques appliquées**

Par  
**Kebbiche Zakia**

## **THEME**

**ETUDE ET EXTENSIONS D'ALGORITHMES DE POINTS  
INTERIEURS POUR LA PROGRAMMATION NON LINEAIRE**

**Soutenue le : 10 / 11 / 2007**

**Devant le jury :**

<b>Président :</b>	<b>Dr. N. Bensalem</b>	<b>Prof. Université Ferhat Abbas-Sétif</b>
<b>Rapporteurs :</b>	<b>Dr. A. Keraghel</b>	<b>M.C. Université Ferhat Abbas-Sétif</b>
	<b>Dr. A.Yassine</b>	<b>Prof. Université du Havre (France)</b>
<b>Examineurs :</b>	<b>Dr. Jean -Rodolphe Roche</b>	<b>Prof. Université de Nancy I (France)</b>
	<b>Dr. Sérigne Gueye</b>	<b>M.C. Université du Havre (France)</b>
	<b>Dr. D. Benterki</b>	<b>M.C. Université Ferhat Abbas-Sétif</b>

## **REMERCIEMENTS**

*Je remercie Monsieur le professeur Abdelkrim Keraghel directeur du laboratoire de mathématiques fondamentales et numériques (LMFN), université Ferhat Abbas, Sétif et le professeur Adnan Yassine directeur du laboratoire de mathématiques appliquées (LMAH), université du Havre, d'avoir accepté de diriger cette thèse, qui par leurs conseils pertinents, leurs remarques et leurs encouragements m'ont permis de la réaliser.*

*J'exprime mon respect à Monsieur le professeur Naceuredinne Bensalem qui m'a fait l'honneur de présider le jury de cette thèse, je le remercie respectueusement.*

*Je tiens aussi à exprimer mes vifs remerciements et respects à Messieurs : Jean -Rodolphe Roche Professeur à l'université de Nancy I (France), Sérigne Gueye Maître de conférences à l'université du Havre( France) et Djamel Benterki Maître de conférences à l'université Ferhat Abbas-Sétif, qui ont bien accepté de juger ce travail et d'en être les examinateurs.*

*Je remercie spécialement Monsieur : Mohammed Achache, Maître de conférences à l'université Ferhat Abbas-Sétif, pour son aide, ses conseils et ses encouragements.*

*Je remercie toute l'équipe d'optimisation du laboratoire de mathématiques fondamentales et numériques de département de mathématiques de Sétif et l'équipe d'optimisation du laboratoire de mathématiques appliquées du Havre pour leurs aides continues et leurs encouragements.*

*Mes remerciements s'adressent également à toute l'équipe administrative des départements de mathématiques de Sétif et du Havre d'avoir mis à notre service tous les moyens disponibles, en particulier le comité et le conseil scientifique d'avoir entrepris avec une grande souplesse les démarches nécessaires pour la soutenance.*

*Enfin, je remercie tous ceux qui m'ont aidé de près ou de loin quelqu'ils soient et d'où qu'ils soient.*

# TABLE DES MATIÈRES

<i>INTRODUCTION GÉNÉRALE</i> .....	1
------------------------------------	---

## *CHAPITRE I*

### *MÉTHODES DE TRAJECTOIRE CENTRALE POUR LE PROBLÈME DE COMPLÉMENTARITÉ LINÉAIRE*

I.1. Préliminaires.....	4
I.2. Programmation mathématique.....	5
I.2.1. Classification d'un programme mathématique.....	6
I.2.2. Qualification des contraintes.....	6
I.2.3. Principaux résultats d'existence.....	7
I.2.4. Conditions d'optimalité.....	7
I.3. Programmation linéaire (PL).....	8
Méthodes de résolution d'un programme linéaire.....	9
1- Méthode du simplexe.....	9
2- Méthodes de points intérieurs.....	9
I.4. Programmation quadratique (PQ).....	11
Méthode de résolution d'un (PQ).....	12
I.5. Complémentarité linéaire.....	12
Transformation d'un programme quadratique convexe en un programme de complémentarité linéaire.....	13
Transformation d'un programme complémentaire linéaire en un programme quadratique convexe.....	15

Méthodes de résolution d'un programme complémentaire linéaire.....	16
a- Méthode de Lemke.....	16
b- Méthode de points intérieurs.....	16
Méthodes de trajectoire centrale.....	16
Problème d'initialisation.....	23
Méthode de trajectoire centrale avec poids.....	23
Aspects numériques.....	27
Méthode de trajectoire centrale non réalisable.....	37
Aspects numériques.....	39
Commentaire.....	40

## *CHAPITRE II*

### *MINIMISATION D'UNE FONCTION CONVEXE SOUS CONTRAINTES LINÉAIRES PAR UNE MÉTHODE DE TRAJECTOIRE CENTRALE AVEC POIDS*

II.1. Description de la méthode classique.....	42
II.2. Méthode de trajectoire centrale avec poids.....	46
Description.....	46
Algorithme.....	47
Calcul de la direction de déplacement.....	48
Convergence.....	49

## *CHAPITRE III*

### *UNE VARIANTE DE TYPE PROJECTIF POUR LA MINIMISATION D'UNE FONCTION CONVEXE SOUS CONTRAINTES LINÉAIRES*

III.1. Méthode de Karmarkar.....	59
Description de l'algorithme de base.....	60
Convergence de l'algorithme.....	62
III.2. Généralisation de l'algorithme de base.....	63
III.3. Extension de la méthode de Karmarkar .....	65
Formulation du problème.....	65
Linéarisation.....	67
Description de l'algorithme.....	69
Etude de la convergence.....	70
 <i>CONCLUSION</i> .....	 74
 <i>BIBLIOGRAPHIE</i> .....	 76

## **Résumé**

Dans cette thèse, nous présentons une étude algorithmique et numérique concernant la méthode de trajectoire centrale appliquée au problème de complémentarité linéaire considéré comme une formulation unificatrice de la programmation linéaire et la programmation quadratique convexe. Puis, nous proposons deux variantes intéressantes, l'une de trajectoire centrale et l'autre de type projectif avec linéarisation, pour minimiser une fonction convexe différentiable sur un polyèdre.

Les algorithmes sont bien définis et les résultats théoriques correspondants sont établis.

### **Mots clés :**

Programmation linéaire, programmation quadratique convexe, complémentarité linéaire, programmation non linéaire, méthodes de points intérieurs, méthode de trajectoire centrale, méthode projective avec linéarisation.

## **Abstract**

In this thesis, we present an algorithmically and numerical study concerning the central path method for linear complementarity problem which is considered as an unifying framework of linear and quadratic programming. Then, we propose two interesting variants namely the central path and the projective with linearization methods for minimizing a convex differentiable function on a polyhedral set. The algorithms are well defined and the corresponding theoretical results are established.

### **Key words:**

Linear programming, convex quadratic programming, linear complementarity problem, nonlinear programming, central path interior point methods, projective with linearization methods.

# *INTRODUCTION GÉNÉRALE*

La résolution d'un problème d'optimisation non linéaire de taille importante reste toujours un challenge malgré tout le succès observé dans ce domaine.

Le relancement des méthodes de points intérieurs depuis 1984 [9] a donné espoir aux spécialistes du domaine pour réaliser des progrès considérables dans ce sens. Ces méthodes ont fait leurs preuves dans le domaine de la programmation linéaire (PL) notamment par leur bonnes propriétés théoriques (complexité polynomiale et convergence superlinéaire) et leur bon comportement numérique. Aussitôt, des variantes sont introduites pour la programmation quadratique et la complémentarité linéaire.

Ceci étant, il faut signaler tout de même que ( sur le plan technique ), ces méthodes présentent des inconvénients d'ordre algorithmique et numérique entre autre : le problème d'initialisation et le coût excessif de l'itération lié aux choix de la direction et du pas de déplacement.

Notre étude est répartie en deux volets :

Le premier est une étude qualitative d'une classe privilégiée de méthodes de points intérieurs, celle de trajectoire centrale. L'effort est concentré sur le problème d'initialisation pour lequel nous proposons deux alternatives : trajectoire centrale avec poids et trajectoire centrale non réalisable. Le modèle d'application étant celui de la complémentarité linéaire monotone qui couvre la programmation linéaire et la programmation quadratique convexe. Nous avons ainsi amélioré nettement le comportement numérique des algorithmes correspondants.

Le deuxième volet concerne l'extension des méthodes de trajectoire centrale et les méthodes projectives pour des problèmes non linéaires en commençant par minimiser une fonction non linéaire convexe différentiable sur un polyèdre.

A ce propos, le mélange des techniques de linéarisation et les ingrédients de l'approche projective ont donné lieu à un algorithme simple et prometteur. Une variante de la méthode de trajectoire centrale avec poids est également à l'ordre du jour, dans les deux cas, les résultats théoriques sont établis.

La thèse est présentée en trois chapitres :

Dans le premier chapitre, nous présentons quelques notions de base sur la programmation mathématique, puis une étude théorique et numérique des méthodes de trajectoire centrale appliquées à un programme complémentaire linéaire comme formulation unificatrice de la programmation linéaire et la programmation quadratique convexe.

Une extension de la méthode de trajectoire centrale avec poids pour la minimisation d'une fonction convexe sous contraintes linéaires est décrite dans le deuxième chapitre.

Dans le dernier chapitre, les techniques de linéarisation sont combinées avec les ingrédients de l'approche projective pour élaborer une méthode simple possédant des atouts d'un comportement numérique.

# Chapter 1

## Méthodes de trajectoire centrale pour le problème de complémentarité linéaire

## 1.1 Préliminaires

- Un ensemble  $D \subseteq \mathbb{R}^n$  est dit convexe si :

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

-  $D$  est dit affine si :

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

-  $D$  est un polyèdre s'il s'écrit comme suit :

$$D = \{x \in \mathbb{R}^n : p_i^t x \leq \alpha_i, i = 1, \dots, m\}$$

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

- Une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est dite convexe sur  $D$  si :

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

-  $f$  est dite strictement convexe si l'inégalité ci-dessus est toujours stricte  $\forall x \neq y, \forall \lambda \in ]0, 1[$ .

-  $f$  est dite affine si :

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

## 1.2 Programmation mathématique

En général, un programme mathématique est défini comme suit :

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

où :  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction continûment différentiable appelée fonction objectif et  $D$  est appelé ensemble des solutions réalisables présenté souvent comme :  $D = U \cap V$ ,

avec :

$$U = \{x \in \mathbb{R}^n : f_i(x) \leq 0, i = 1, \dots, m\} \text{ et } V = \{x \in \mathbb{R}^n : g_i(x) = 0, i = 1, \dots, p\}.$$

### Définitions

- On appelle solution réalisable de  $(PM)$  tout point vérifiant les contraintes ( i.e. : appartenant à  $D$  ).
- Une solution réalisable qui minimise l'objectif sur  $D$  est dite solution optimale globale de  $(PM)$ . On note par  $\arg \min_D f(x)$  l'ensemble des solutions optimales globales.
- Un point  $x^* \in D$  est une solution optimale locale de  $(PM)$  si :  $\exists V$  (voisinage) de  $x^*$  tel que :  $f(x^*) \leq f(x), \forall x \in V$  et on note par  $\text{loc} \min_D f(x)$  l'ensemble des solutions optimales locales de  $(PM)$ .

Nous avons toujours  $\arg \min f(x) \subseteq \text{loc} \min f(x)$  et si  $(PM)$  est convexe les deux ensembles sont égaux.

### Remarque 1

Le problème d'optimisation précédent consiste :

- Soit à chercher un point optimal ( local ou global ).
- Soit, si un tel point n'existe pas on cherche une borne inférieure à  $f$ .
- Soit à établir que  $f$  est non bornée inférieurement sur  $D$ , auquel cas on adopte la convention  $\inf_D f(x) = -\infty$ .
- Lorsque  $D$  est vide on pose par convention  $\inf_D f(x) = +\infty$ .

### 1.2.1 Classification d'un programme mathématique

On classifie le problème  $(PM)$  à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité des fonctions du problème. A ce propos,  $(PM)$  est dit convexe si  $f$  et  $f_i$  sont convexes et  $g_i$  affines. Si ces dernières sont toutes différentiables, on dit que  $(PM)$  est différentiable. La classe de programmes mathématiques convexes différentiables est le modèle le mieux élaboré, les programmes non convexes ou non différentiables sont difficiles à traiter.

Enfin, le cas le plus simple est celui de la programmation linéaire où  $f, f_i$  et  $g_i$  sont affines.

### 1.2.2 Qualification des contraintes

- Si  $D$  est un polyèdre convexe ( i.e. : toutes les fonctions contraintes sont affines ), alors par définition les contraintes sont qualifiées en tout point réalisable.
- Si  $D$  est convexe ( i.e. :  $f_i$  convexe et  $g_i$  affine ) et  $intD \neq \emptyset$ , les contraintes sont qualifiées partout. C'est la condition de Slater.
- Une contrainte d'inégalité  $f_i(x) \leq 0$  est dite saturée (ou active) en  $x^* \in D$  si  $f_i(x^*) = 0$ . De ce fait, une contrainte d'égalité  $g_i(x) = 0$  est par définition saturée en tout point  $x$  de  $D$ . les contraintes sont qualifiées en  $x^* \in D$  si les gradients de toutes les fonctions des contraintes saturées en  $x^*$  sont linéairement indépendants.

#### Remarque 2

La résolution complète de  $(PM)$  traite dans l'ordre les points suivants :

- L'existence ( et éventuellement l'unicité ) d'une solution optimale.
- La caractérisation de la solution (il s'agit des conditions d'optimalité ).
- L'élaboration d'algorithmes pour calculer cette solution.

### 1.2.3 Principaux résultats d'existence

#### Théorème 1 ( Weierstrass ) [4]

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

#### Corollaire 1

Soit  $D$  fermé non vide de  $\mathbb{R}^n$ ,  $f$  est continue et coercive sur  $D$  ( au sens que  $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$  ) alors (PM) admet au moins une solution optimale globale.  $\square$

#### Remarque 3

L'unicité d'une solution éventuelle est en général une conséquence de la stricte convexité de  $f$ .

### 1.2.4 Conditions d'optimalité

On considère le programme mathématique :

$$(PM) \begin{cases} \min f(x) \\ f_i(x) \leq 0, \quad i = 1, \dots, m \\ g_i(x) = 0, \quad i = 1, \dots, p \end{cases}$$

où  $f, f_i$  et  $g_i$  sont continûment différentiables.

#### Théorème 2 ( Karush-Kuhn-Tucher (KKT) ) [19]

Si  $x^*$  est une solution optimale locale de (P) satisfaisant l'une des conditions de qualification précédentes, alors il existe des multiplicateurs  $\lambda \in \mathbb{R}_+^m$  et  $\mu \in \mathbb{R}^p$  tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla f_i(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0 & ( \text{condition d'optimalité} ) \\ \lambda_i f_i(x^*) = 0, \quad i = 1, \dots, m & ( \text{condition de complémentarité} ) \\ g_i(x^*) = 0, \quad i = 1, \dots, p \end{cases}$$

#### Remarque 4

- Si les contraintes ne sont pas qualifiées en  $x^*$ , les conditions de KKT ne s'appliquent pas (  $x^*$  peut être optimal sans vérifier ces conditions ).
- Si  $(PM)$  est convexe, les conditions de KKT sont à la fois nécessaires et suffisantes pour que  $x^*$  soit un minimum global.

### 1.3 Programmation linéaire (PL)

On considère le programme linéaire sous la forme standard :

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

où  $A$  est une matrice réelle de type  $(m, n)$  supposée de plein rang ( $rgA = m < n$ ),  $b \in \mathbb{R}^m$  et  $c \in \mathbb{R}^n$ .

L'ensemble des solutions réalisables  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  est un polyèdre convexe fermé.

Le dual de  $(PL)$  est donné par :

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

#### Proposition 1 [19]

*Un programme linéaire réalisable et borné ( objectif borné ) possède au moins une solution optimale, située sur la frontière du domaine réalisable.  $\square$*

#### Définitions

- Une base de  $A$  est une sous matrice  $B$  formée de  $m$  colonnes linéairement indépendantes de  $A$ .

- La solution de base associée à  $B$  est le point  $x = (x_B, x_N) \in \mathbb{R}^n$  tel que :  $x_B = B^{-1}b, x_N = 0$ .
- Une solution de base qui vérifie  $x_B \geq 0$  est dite solution de base réalisable.
- Un point  $x$  est un sommet de  $S$  si et seulement s'il est une solution de base réalisable.
- Un sommet est dit non dégénéré si  $x_B > 0$ . Il est dégénéré dans le cas contraire ( au moins une composante de  $x_B$  est nulle).

**Proposition 2 [19]**

*Si un programme linéaire possède une solution optimale, alors au moins un sommet du domaine réalisable est une solution optimale. □*

## Méthodes de résolution d'un programme linéaire

### 1) Méthode du simplexe

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

Dans le cas dégénéré l'algorithme risque de cycler, cependant il existe des techniques convenables pour éviter ce phénomène. En général, la méthode du simplexe possède un comportement numérique très satisfaisant confirmé par ses applications multiples dans la résolution d'une large classe de problèmes pratiques. En théorie, la méthode n'a pas autant de succès, elle est plutôt jugée inefficace de par sa complexité arithmétique exponentielle de l'ordre de  $O(2^n)$  opérations.

### 2) Méthodes de points intérieurs

Ces méthodes ont été introduites à la fin des années 50 dans le but de résoudre des programmes mathématiques non linéaires. Leur utilisation pour la programmation linéaire n'a pas reçu autant d'enthousiasme à cause de la dominance quasi-totale de la

méthode du simplexe à cette époque. Après l'apparition de l'algorithme de Karmarkar en 1984 pour la programmation linéaire, les méthodes de points intérieurs ont connu une véritable révolution, on enregistre plus de 3000 publications en quelques années.

On distingue trois classes fondamentales de méthodes de points intérieurs à savoir : les méthodes affines, les méthodes de réduction du potentiel et les méthodes de trajectoire centrale ( chemin central ).

#### **a/ Méthodes affines (Dikin 1967) :**

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

L'algorithme est d'une structure simple, malheureusement, il n'est pas facile de démontrer la polynômialité.

#### **b/ Méthodes de réduction du potentiel :**

La fonction potentiel joue un grand rôle dans le développement des méthodes de points intérieurs. En effet, une réduction de cette fonction conduit directement à celle de la fonction objectif. L'algorithme de Karmarkar appliqué au programme linéaire sous forme standard utilise une fonction potentiel de la forme :  $n \ln(c^t x - Z) - \sum_{i=1}^n \ln(x_i)$ , où  $Z$  est une borne inférieure de la valeur optimale de l'objectif. Karmarkar prouve la convergence et la polynômialité de son algorithme par montrer que cette fonction est réduite à chaque itération par au moins une constante. Depuis 1987, les chercheurs introduisent des fonctions du potentiel de type primal-dual, parmi lesquelles, celle de Tanabe, Todd et Ye [24] définie par :  $\Phi_\rho(x, s) = \rho \ln(x^t s) - \sum_{i=1}^n \ln(x_i s_i)$ , pour  $\rho > n$ . Cette fonction a joué un rôle très important dans le développement des algorithmes de réduction du potentiel après 1988. Les algorithmes correspondants à ces méthodes possèdent une complexité polynômiale, ils nécessitent  $O(\sqrt{n} |\ln \varepsilon|)$  itérations pour réduire le saut de dualité (  $x^t s \leq \varepsilon$ ,  $\varepsilon$  est une précision donnée ).

### c/ Méthodes de trajectoire centrale (TC) :

Elles ont été introduites à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles possèdent de bonnes propriétés théoriques : une complexité polynômiale et une convergence superlinéaire.

Les algorithmes de (TC) restreignent les itérés à un voisinage du chemin central, ce dernier est un arc de points strictement réalisables.

Plusieurs chercheurs essaient de généraliser le principe de ces méthodes pour la programmation non linéaire. En 1987, Megiddo [24] a proposé un algorithme primal-dual de trajectoire centrale pour la programmation linéaire avec une généralisation pour le problème de complémentarité linéaire (*PCL*). Kojima et al. [15] ont développé un algorithme primal-dual pour la programmation linéaire. Une extension pour le (*PCL*) est proposée par les mêmes chercheurs en 1989 avec la complexité  $O(\sqrt{n} \ln(\frac{1}{\varepsilon}))$  itérations.

## 1.4 Programmation quadratique

La programmation quadratique est connue pour ses applications multiples dans plusieurs domaines. Souvent, elles interviennent comme procédures intermédiaires pour des programmes non linéaires, c'est le cas entre autre des méthodes de programmation quadratique successives (SQP).

Sans perte de généralité, on peut présenter un programme quadratique sous la forme suivante :

$$(PQ) \left\{ \begin{array}{l} \min f(x) = \left( \frac{1}{2} x^t Q x + c^t x \right) \\ Ax = b \\ x \geq 0 \end{array} \right.$$

où  $Q$  est une matrice symétrique d'ordre  $n$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$  et  $A \in \mathbb{R}^{m \times n}$  de plein rang ( $rgA = m$ ).

Rappelons que l'ensemble des contraintes  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  est un polyèdre convexe et fermé, la fonction objectif est infiniment différentiable.

$(PQ)$  est convexe si et seulement si  $f$  est convexe auquel cas la matrice  $Q$  est semi-définie positive.

### Méthodes de résolution d'un (PQ)

On peut classifier les méthodes de résolution en deux catégories conformément à leurs principes : les méthodes simpliciales et les méthodes de points intérieurs qui sont des extensions des algorithmes proposés pour le cas linéaire.

## 1.5 Complémentarité linéaire

Depuis son apparition, la complémentarité a attiré l'intérêt de plusieurs chercheurs, son importance peut être mesurée par le rôle crucial qu'elle joue dans la résolution de plusieurs problèmes dans différents domaines : programmation linéaire, programmation quadratique convexe, inéquations variationnelles, mécanique, ...

Le problème de complémentarité linéaire noté  $(PCL)$  généralise la programmation linéaire et la programmation quadratique convexe.

Il s'écrit sous la forme :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } x, y \in \mathbb{R}^n \text{ tels que :} \\ y = Mx + q \\ x^t y = 0 \\ (x, y) \geq 0 \end{array} \right.$$

où :  $M$  est une matrice carrée d'ordre  $n$  et  $q \in \mathbb{R}^n$  .

## Transformation d'un programme quadratique convexe en un problème complémentaire linéaire

Soit le programme quadratique convexe :

$$(PQC) \begin{cases} \min \frac{1}{2}x^tQx + c^tx \\ Ax \leq b \\ x \geq 0 \end{cases}$$

où  $Q$  est une matrice carrée d'ordre  $n$ , symétrique et semi-définie positive.

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

Comme le problème  $(PQC)$  est convexe et les contraintes sont linéaires alors les conditions de KKT sont nécessaires et suffisantes et s'écrivent comme suit :

$x \in \mathbb{R}_+^n$  est une solution optimale de  $(PQC)$  ssi  $\exists y \in \mathbb{R}^m, \lambda \in \mathbb{R}^n$  tels que :

$$\begin{cases} c + Qx + A^ty - \lambda = 0 \\ y^t(b - Ax) = 0 \\ -\lambda^tx = 0 \\ y \geq 0, \lambda \geq 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \lambda = c + A^ty + Qx \\ v = b - Ax \\ \lambda^tx = 0, y^tv = 0 \\ x \geq 0, y \geq 0, \lambda \geq 0, v \geq 0 \end{cases}$$

$$\Leftrightarrow \left\{ \begin{array}{l} \begin{pmatrix} \lambda \\ v \end{pmatrix} = \begin{bmatrix} Q & A^t \\ -A & 0 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ b \end{pmatrix} \\ \left\langle \begin{pmatrix} \lambda \\ v \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right\rangle = 0 \\ \begin{pmatrix} \lambda \\ v \end{pmatrix} \geq 0, \begin{pmatrix} x \\ y \end{pmatrix} \geq 0 \end{array} \right.$$

En posant :

$$w = \begin{pmatrix} \lambda \\ v \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, q = \begin{pmatrix} c \\ b \end{pmatrix}$$

$$M = \begin{bmatrix} Q & A^t \\ -A & 0 \end{bmatrix}$$

On obtient :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ w = Mz + q \geq 0, z \geq 0 \\ w^t z = 0 \end{array} \right.$$

### Remarque 5

Tout programme linéaire sous la forme :

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

peut se transformer à un  $(PCL)$  de la forme :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ w = Mz + q \geq 0, z \geq 0 \\ w^t z = 0 \end{array} \right.$$

où:

$$w = \begin{pmatrix} \lambda \\ v \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, q = \begin{pmatrix} c \\ b \end{pmatrix}$$
$$M = \begin{bmatrix} 0 & A^t \\ -A & 0 \end{bmatrix}$$

## Transformation d'un problème complémentaire linéaire monotone en un programme quadratique convexe

En général, on ne peut pas transformer un  $(PCL)$  quelconque en  $(PQC)$  sauf si la matrice  $M$  est semi-définie positive auquel cas on parle de  $(PCL)$  monotone.

En effet, soit le problème complémentaire linéaire monotone suivant :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ w = Mz + q \geq 0, z \geq 0 \\ w^t z = 0 \end{array} \right.$$

On a le théorème suivant :

### **Théorème 3 [24]**

*$(PCL)$  est équivalent au programme quadratique convexe suivant :*

$$\left\{ \begin{array}{l} \min z^t(Mz + q) \\ Mz + q \geq 0 \\ z \geq 0 \end{array} \right.$$

*au sens que  $(z^*, Mz^* + q)$  est une solution de  $(PCL)$  ssi  $z^*$  est une solution optimale du programme quadratique avec une valeur optimale nulle de l'objectif.  $\square$*

## Méthodes de résolution d'un programme complémentaire linéaire

Les méthodes existantes pour résoudre (*PCL*) sont des extensions des méthodes conçues pour la programmation linéaire.

### 1) Méthode de Lemke (1969) [4]

c'est une variante de la méthode du simplexe basée sur la notion de pivots complémentaires ( la variable qui entre en base et celle qui quitte la base doivent vérifier la condition de complémentarité ). L'algorithme de Lemke est connu pour son bon comportement numérique, alors qu'en théorie, il est jugé inefficace à cause de sa complexité arithmétique exponentielle exprimée par le nombre total de pivots éventuels.

### 2) Méthodes de points intérieurs

Le succès observé dans les méthodes de points intérieurs pour la programmation linéaire a incité les chercheurs à développer des extensions intéressantes pour résoudre le problème de complémentarité linéaire, entre autre les méthodes de trajectoire centrale à cause de leurs propriétés attractives, notamment la complexité polynômiale et la convergence superlinéaire ce qui devrait donner lieu à un comportement numérique prometteur.

## Méthodes de trajectoire centrale

Cette approche est proposée par Kojima, Mizuno et Yoshise [15], c'est une extension de la méthode de trajectoire centrale appliquée à la programmation linéaire.

Soit le problème complémentaire linéaire suivant :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } x \in \mathbb{R}^n \text{ tel que :} \\ y = Mx + q \\ x^t y = 0 \\ (x, y) \geq 0 \end{array} \right.$$

Notons par :

$S = \{(x, y) \in \mathbb{R}_+^{2n} : y = Mx + q\}$  l'ensemble des solutions réalisables de (*PCL*).

$S_{int} = \{(x, y) \in \mathbb{R}_{++}^{2n} : y = Mx + q\}$  l'ensemble des solutions strictement réalisables de  $(PCL)$ .

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

$$X = \text{diag}(x) : x > 0, Xe = x \text{ et } X^{-1}e = \text{diag}\left(\frac{1}{x}\right).e$$

$$Y = \text{diag}(y) : y > 0, Ye = y \text{ et } Y^{-1}e = \text{diag}\left(\frac{1}{y}\right).e$$

$$X^{-2} = \text{diag}\left(\frac{1}{x^2}\right).$$

$X, X^{-1}, X^{-2}, Y$  et  $Y^{-1}$  : sont des matrices diagonales, symétriques et définies positives.

Sans perte de généralité, on suppose que :

**(H1)**  $S_{int} \neq \emptyset$ , c'est à dire, un point strictement réalisable existe.

**(H2)**  $M$  est semi-définie positive.

Le  $(PCL)$  monotone est équivalent au programme quadratique convexe suivant :

$$\begin{cases} \min x^t y \\ y = Mx + q \\ (x, y) \geq 0 \end{cases}$$

La notion de trajectoire centrale peut être introduite moyennant une fonction barrière logarithmique, il suffit d'associer à  $(PCL)$  le problème barrière suivant :

$$(PCL)_\mu \begin{cases} \min f_\mu(x, y) = \left[ x^t y - \mu \sum_{i=1}^n \ln(x_i y_i) \right], \mu > 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

Si  $\mu \rightarrow 0$ , la solution de  $(PCL)_\mu$  converge vers la solution de  $(PCL)$ , donc on résout une suite de problèmes  $(PCL)_\mu$  en diminuant les valeurs de  $\mu$  jusqu'à l'obtention d'une solution de  $(PCL)$ .

L'idée générale des méthodes de trajectoire centrale consiste à suivre un chemin particulier (dit des centres ou chemin central), en prenant comme direction de déplacement celle de Newton. Autrement dit : l'algorithme génère une suite  $(x^k, y^k)$  strictement réal-

isable et une suite  $(\mu^k)$  qui exprime la condition de complémentarité. Cette dernière est vérifiée lorsque  $(\mu^k)$  tend vers zéro.

**Proposition 3**

*La fonction  $f_\mu(x, y)$  est strictement convexe.*

**Preuve**

La fonction  $f_\mu(x, y)$  peut être écrite sous la forme :

$$f_\mu(x) = x^t(Mx + q) - \mu \left[ \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \ln(Mx + q)_i \right]$$

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

$$\nabla f_\mu(x) = (M + M^t)x + q - \mu X^{-1}e - \mu M^t \text{diag} [(Mx + q)_1, \dots, (Mx + q)_n]^{-1} e$$

$$\nabla^2 f_\mu(x) = M + M^t + \mu [X^{-2} + M^t \text{diag} ((Mx + q)_1, \dots, (Mx + q)_n)^{-2} M].$$

Comme  $M$  est une matrice semi-définie positive,  $X^{-2}$  est définie positive pour tout  $x > 0$  et  $\mu > 0$ , alors  $\nabla^2 f_\mu(x)$  est une matrice définie positive, d'où le résultat.  $\square$

**Théorème 4 [15]**

*$M$  étant semi-définie positive et  $S_{int} \neq \emptyset$ , le problème  $(PCL)_\mu$  admet une solution optimale unique pour tout  $\mu > 0$ .  $\square$*

**Théorème 5**

*Soient  $\mu > 0$  et  $M$  une matrice semi-définie positive.*

*$(x, y)$  est une solution optimale de  $(PCL)_\mu$  si et seulement si  $(x, y)$  satisfait le système suivant :*

$$(*) \left\{ \begin{array}{l} XYe - \mu e = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

*Donc, la résolution de  $(PCL)_\mu$  est équivalente à résoudre le système  $(*)$ .*

### Preuve

$(PCL)_\mu$  est convexe et différentiable, les contraintes sont qualifiées car affines, alors les conditions de KKT sont nécessaires et suffisantes et s'écrivent comme suit :

$$\left\{ \begin{array}{l} y - \mu X^{-1}e + M^t v = 0 \\ x - \mu Y^{-1}e - v = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

ou d'une manière équivalente :

$$\left\{ \begin{array}{l} (XM^t + Y)v = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

$(XM^t + Y)$  est définie positive alors inversible, donc,  $v = 0$ . On substitut dans le système de KKT, on obtient :

$$\left\{ \begin{array}{l} y - \mu X^{-1}e = 0 \\ x - \mu Y^{-1}e = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

Alors:

$$\left\{ \begin{array}{l} XYe - \mu e = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

D'où le résultat.  $\square$

### Définition

En désignant par  $(x_\mu, y_\mu)$  la solution du système non linéaire (\*) pour  $\mu > 0$  donné.

L'ensemble de toutes les solutions du système (\*):  $T = \{(x_\mu, y_\mu) : \mu > 0\}$  est appelé trajectoire centrale.

### Résolution du système (\*)

Puisque la première équation du système (\*) est non linéaire, une résolution directe est généralement impossible. Alors, on cherche une solution approchée  $(x_\mu, y_\mu)$  au système d'équations suivant :  $F_\mu(x, y) = 0$  et  $(x, y, \mu) \in \mathbb{R}_+^{2n+1}$  .....(\*)

où  $F_\mu : \mathbb{R}_+^{2n} \rightarrow \mathbb{R}^{2n}$  est une fonction définie par :

$$F_\mu(x, y) = \begin{bmatrix} XYe - \mu e \\ y - Mx - q \end{bmatrix}$$

On note que :  $(x, y)$  est une solution de (PCL) si et seulement si elle est une solution du système (\*) pour  $\mu = 0$ .

On applique la méthode de Newton au système (\*) à partir d'un point réalisable  $w = (x, y)$ , on obtient :

$$F_\mu(w) + \nabla F_\mu(w) \cdot \Delta w = 0, \text{ où } \Delta w = (\Delta x, \Delta y)$$

C'est à dire, le système d'équations linéaires suivant :

$$\begin{cases} Y\Delta x + X\Delta y = \mu e - XYe \\ \Delta y = M\Delta x \end{cases}$$

dont la solution  $(\Delta x, \Delta y)$  est donnée par :

$$\begin{cases} \Delta x = (M + X^{-1}Y)^{-1} (\mu X^{-1}e - Y e) \\ \Delta y = M\Delta x \end{cases}$$

Le nouvel itéré est calculé comme suit :  $(\bar{x}, \bar{y}) = (x, y) + (\Delta x, \Delta y)$ .

## Facteur de centralité

La qualité de chaque solution trouvée est mesurée par un facteur dit de centralité : c'est le scalaire défini par :

$$\delta(x, y, \mu) = \min_{\mu \in \mathbb{R}_+} \|XYe - \mu e\| = \left\| XYe - \left(\frac{x^t y}{n}\right)e \right\|$$

$(x, y) \in S_{int}$  est un point de la trajectoire centrale si :  $\left\| XYe - \left(\frac{x^t y}{n}\right)e \right\| = 0$ .

On veut contrôler notre approximation de la trajectoire telle que la déviation  $\left\| XYe - \left(\frac{x^t y}{n}\right)e \right\|$  converge vers zéro au moins linéairement lorsque le terme  $x^t y$  tend vers zéro. Ceci nous conduit à définir le voisinage de la trajectoire centrale noté  $T(\theta)$ . On dit qu'un point est voisin de la trajectoire centrale s'il appartient à l'ensemble suivant :

$$T(\theta) = \left\{ (x, y) \in S_{int} : \left\| XYe - \left(\frac{x^t y}{n}\right)e \right\| \leq \left(\frac{x^t y}{n}\right)\theta : \theta > 0 \right\}$$

## Description algorithmique :

Partant d'une solution initiale  $(x^0, y^0)$  dans le voisinage de la trajectoire centrale avec  $\mu = \mu^0$ , l'algorithme génère une suite de point  $(x^k, y^k)$  solution du système  $(\tilde{*})$  par la méthode de Newton. On s'arrête lorsqu'on obtient une valeur suffisamment petite du paramètre  $\mu$ .

Pour des choix convenables des constantes positives  $\beta$  et  $\theta$  et le paramètre  $\mu$ , l'algorithme génère une suite  $(x^k, y^k) \in T(\theta)$  ( tous les itérés restent dans le voisinage de la trajectoire centrale ) en partant d'un point initial  $(x^0, y^0) \in T(\theta)$ , de plus, à chaque itération le terme  $(x^k)^t y^k$  se réduit au moins linéairement. L'algorithme correspondant se présente comme suit :

## Algorithme de base

### Début algorithme

**Données :**  $\varepsilon > 0$  un paramètre de précision,  $\beta \in (0, 1)$  et  $0 < \theta \leq 0.1$  une constante donnée.

**Initialisation :**  $(x^0, y^0) \in T(\theta)$

**Itération :** pour  $k = 0, 1, 2, \dots$

Calculer :

- Le paramètre :  $\mu^k = \frac{(x^k)^t y^k}{n}$ .

- La direction de Newton  $(\Delta x^k, \Delta y^k)$  solution du système linéaire :

$$\begin{cases} Y^k \Delta x^k + X^k \Delta y^k = \beta \mu^k e - X^k Y^k e \\ \Delta y^k = M \Delta x^k \end{cases}$$

- Le nouvel itéré :  $(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\Delta x^k, \Delta y^k)$ ,  $k = k + 1$ .

**Test d'arrêt :**  $(x^k)^t y^k \leq \varepsilon$

**Fin algorithme.**

### Convergence

Si le point courant  $(x^k, y^k)$  est voisin de la trajectoire centrale et si on choisit une valeur convenable du paramètre  $\mu^k$ , alors le nouveau point  $(x^{k+1}, y^{k+1})$  reste voisin de la trajectoire centrale, plus précisément, on a le théorème suivant :

#### **Théorème 6 [15]**

Soient :  $0 < \theta \leq 0.1$ ,  $\delta = \frac{\theta}{1 - \theta}$  et  $\beta = 1 - \frac{\delta}{\sqrt{n}}$ , supposons que :

$(x^k, y^k) \in T(\theta)$  et  $\mu^k = \frac{(x^k)^t y^k}{n}$ , alors le point  $(x^{k+1}, y^{k+1})$  défini par :

$(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\Delta x^k, \Delta y^k)$  satisfait :

1/  $(x^{k+1}, y^{k+1}) \in T(\theta)$

2/  $(x^{k+1})^t y^{k+1} \leq (1 - \frac{\delta}{6\sqrt{n}})(x^k)^t y^k$ .  $\square$

## Problème d'initialisation

En pratique, trouver une solution strictement réalisable initiale qui soit proche de la trajectoire centrale est une difficulté majeure faisant l'objet de recherche. Pour remédier à cette difficulté, nous présentons deux alternatives différentes : la méthode de trajectoire centrale avec poids et la méthode de trajectoire centrale non réalisable.

## Méthode de trajectoire centrale avec poids

On considère le problème perturbé suivant :

$$(PCL)_{\mu r} \left\{ \begin{array}{l} \min f_{\mu r}(x, y) = \left[ x^t y - \mu \sum_{i=1}^n r_i \ln(x_i y_i) \right], \mu > 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

où  $r = (r_1, r_2, \dots, r_n)^t \in \mathbb{R}_{++}^n$  est le poids associé à la fonction barrière logarithmique.

Notons que  $(PCL)_{\mu r}$  généralise  $(PCL)_{\mu}$  au sens que si  $r_i = 1, \forall i = 1, \dots, n$ , on retrouve ce dernier. La solution optimale de  $(PCL)_{\mu r}$  converge vers la solution de (PCL) lorsque  $\mu \rightarrow 0$ .

### **Théorème 7 [7]**

*Pour chaque  $\mu > 0$ , le problème  $(PCL)_{\mu r}$  possède une solution optimale unique.  $\square$*

Comme dans le cas classique, la résolution de  $(PCL)_{\mu r}$  est équivalente à celle du système non linéaire suivant :

$$(**) \left\{ \begin{array}{l} XYe - \mu r = 0 \\ y = Mx + q \\ (x, y) > 0 \end{array} \right.$$

L'ensemble de toutes les solutions du système (\*\*):  $T_r = \{(x_{\mu}, y_{\mu}) : \mu > 0\}$  est appelé trajectoire centrale.

Le facteur de centralité est le scalaire défini par :

$$\delta(x, y, \mu) = \min_{\mu \in \mathbb{R}_+} \|XYe - \mu r\| = \left\| XYe - \left(\frac{x^t Ry}{n}\right)r \right\|$$

On dit qu'un point est voisin de la trajectoire centrale s'il appartient à l'ensemble suivant :

$$T_r(\theta) = \left\{ (x, y) \in S_{int} : \left\| XYe - \left(\frac{x^t Ry}{n}\right)r \right\| \leq \left(\frac{x^t Ry}{n}\right)\theta : \theta > 0 \right\}$$

Pour résoudre le système (\*\*), on définit la fonction  $F_{\mu r} : \mathbb{R}_+^{2n} \rightarrow \mathbb{R}^{2n}$  par :

$$F_{\mu r}(x, y) = \begin{bmatrix} XYe - \mu r \\ y - Mx - q \end{bmatrix}$$

La résolution de  $F_{\mu r}(x, y) = 0$  par la méthode de Newton revient à résoudre le système linéaire suivant :

$$\begin{cases} Y\Delta x + X\Delta y = \mu r - XYe \\ \Delta y = M\Delta x \end{cases}$$

Par des simples calculs, on obtient :

$$\begin{cases} \Delta x = (M + X^{-1}Y)^{-1} (\mu X^{-1}r - Y e) \\ \Delta y = M\Delta x \end{cases}$$

Le nouvel itéré est calculé comme suit :  $(\bar{x}, \bar{y}) = (x, y) + (\Delta x, \Delta y)$ .

### Description de l'algorithme

Soit  $(x^0, y^0) \in S_{int}$  ( point strictement réalisable ), on pose :  $r = \frac{X^0 Y^0 e}{\mu^0}$  avec  $\mu^0 = \frac{\|X^0 Y^0 e\|}{\sqrt{n}}$ , le poids  $r$  est choisi de telle manière que  $(x^0, y^0)$  soit un point de la trajectoire centrale, c'est à dire,  $\|X^0 Y^0 e - \mu^0 r\| = 0$  et alors  $(x^0, y^0) \in T_r(\theta)$ .

Partant donc du point initial  $(x^0, y^0) \in T_r(\theta)$ , pour des choix convenables de  $\beta$  et le paramètre  $\mu$ , l'algorithme génère une suite de points  $(x^k, y^k) \in T_r(\theta)$ , de plus, à chaque itération de l'algorithme, le terme  $(x^k)^t R y^k$  se réduit au moins linéairement. L'algorithme basé sur cette méthode se présente comme suit :

## Algorithme

### Début algorithme

**Données :** Soient  $\varepsilon > 0$  un paramètre de précision et  $\beta \in (0, 1)$ .

**Initialisation :**  $(x^0, y^0) \in S_{int}, \mu = \frac{\|X^0 Y^0 e\|}{\sqrt{n}}$  et  $r = \frac{X^0 Y^0 e}{\mu}$ .

$X^0 = \text{diag}(x^0), Y^0 = \text{diag}(y^0), R = \text{diag}(r)$  et  $\eta = \min r_i$ , pour  $i = 1, \dots, n$ .

**Itération :** Pour  $k = 0, 1, 2, \dots$

Calculer :

- Le paramètre :  $\mu^k = \frac{(x^k)^t R y^k}{n}$

- La direction de Newton  $(\Delta x^k, \Delta y^k)$  solution du système linéaire :

$$\begin{cases} Y^k \Delta x^k + X^k \Delta y^k = \beta \mu^k r - X^k Y^k e \\ \Delta y^k = M \Delta x^k \end{cases}$$

- Le nouvel itéré :  $(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\Delta x^k, \Delta y^k), k = k + 1$ .

**Test d'arrêt :**  $(x^k)^t R y^k \leq \varepsilon$

**Fin algorithme.**

### Convergence

On a le théorème suivant :

#### Théorème 8 [7]

Soient  $(x^0, y^0) \in S_{int}, \mu = \frac{\|X^0 Y^0 e\|}{\sqrt{n}}, r = \frac{X^0 Y^0 e}{\mu}, \theta = \delta = \left(1 - \frac{\sqrt{2}}{2}\right) \eta$  et  $\beta = 1 - \frac{\delta}{\sqrt{n}}$ .

$\eta = \min r_i$ , pour  $i = 1, \dots, n$ .

Supposons que :  $(x^k, y^k) \in T_r(\theta)$  et  $\mu^k = \frac{(x^k)^t R y^k}{n}$ , alors :

1)  $(x^{k+1}, y^{k+1}) \in T_r(\theta)$

2)  $(x^{k+1})^t R y^{k+1} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) (x^k)^t R y^k$

où :  $T_r(\theta) = \{(x, y) \in S_{int} : \|XYe - \mu r\| \leq \mu \theta : \theta > 0\}$ .  $\square$

#### Remarque 6

Notons que l'introduction du poids entraîne une grande flexibilité dans le choix du point initial sans détériorer les résultats théoriques.

## Phase d'initialisation

**Pas 1)** Pour trouver un point initial strictement réalisable, on résout le système suivant :

$$(P1) \begin{cases} y = Mx + q \\ (x, y) > 0 \end{cases}$$

D'après Karmarkar [10], (P1) est équivalent au problème d'optimisation :

$$(P2) \begin{cases} \min \lambda \\ Az + \lambda(q - Az^0) = q \\ \lambda \geq 0, z \geq 0 \end{cases}$$

où  $z^0 > 0$  est choisi arbitrairement dans l'orthant positif et  $\lambda$  une variable artificielle.

$$A = [I \quad -M] \text{ et } z = \begin{pmatrix} y \\ x \end{pmatrix}.$$

Notons que (P2) est toujours réalisable, il suffit de prendre :  $z = z^0$  et  $\lambda = 1$ .

Karmarkar démontre le théorème suivant :

### **Théorème 9 [13]**

$\exists \varepsilon_0 > 0$  tel que les deux propositions suivantes sont équivalentes :

- 1) (P1) est réalisable.
- 2) (P2) admet une solution optimale  $(z, \lambda)$  telle que  $\lambda \leq \varepsilon_0$ .

La résolution de (P1) se ramène donc à celle de (P2), problème auquel on appliquera l'algorithme de Karmarkar, puisque connaissant la valeur optimale et une solution réalisable de ce dernier. ( Voir chapitre 3 ).

**Pas 2)** On introduit le poids  $r$  en posant :  $r = \frac{X^0 Y^0 e}{\mu}$  avec  $\mu = \frac{\|X^0 Y^0 e\|}{\sqrt{n}}$  donc on se retrouve avec  $(x^0, y^0)$  un point de la trajectoire centrale.

## Aspects numériques

Nous présentons dans ce paragraphe quelques expérimentations numériques issues de la mise en oeuvre de l'algorithme de trajectoire centrale avec poids décrit précédemment. L'algorithme est programmé en Turbo-Pascal. Les tests sont effectués sur Pentium 4 avec une précision de  $10^{-6}$ .

Soient les programmes linéaires suivants :

### Exemple 1

$$\left\{ \begin{array}{l} \min -4x_1 - 5x_2 \\ 2x_1 + x_2 \leq 8 \\ x_1 + 2x_2 \leq 7 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{array} \right.$$

La solution optimale est :  $x^* = (3, 2)^t$ .

### Exemple 2

$$\left\{ \begin{array}{l} \min 3x_1 - x_2 + x_3 \\ 2x_1 + x_2 - x_4 \leq 0 \\ x_3 + x_5 - x_6 \leq 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{array} \right.$$

La solution optimale est :  $x^* = (0, 0.5, 0, 0.5, 0, 0)^t$ .

### Exemple 3

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

La matrice  $A$ , les vecteurs  $b$  et  $c$  sont présentés dans le tableau suivant :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 5 & 4 & 3 & 2 & 1 \\ 6 & 7 & 8 & 9 & 10 & 5 & 2 & 8 & 3 & 1 \\ 11 & 12 & 13 & 14 & 15 & 6 & 7 & 80 & 90 & 10 \\ 1 & 10 & 20 & 30 & 40 & 50 & 60 & 80 & 90 & 10 \\ 3 & 9 & 27 & 60 & 45 & 60 & 75 & 8 & 9 & 46 \end{bmatrix}, b = \begin{pmatrix} 10000 \\ 10000 \\ 10000 \\ 10000 \\ 10000 \end{pmatrix}$$

$$c_i = -1, i = 1, \dots, 10.$$

La solution optimale est :  $x^* = (841.121495, 0, 0, 0, 0, 124.610592, 0, 0, 0, 0)^t$ .

Les résultats obtenus sont présentés dans le tableau suivant :

	Nombre d'itérations	Temps de calcul (s)
<b>Exemple 1</b>	<b>141</b>	<b>0.06</b>
<b>Exemple 2</b>	<b>3628</b>	<b>1.97</b>
<b>Exemple 3</b>	<b>27946</b>	<b>59.54</b>

Les solutions initiales obtenues par l'algorithme de Karmarkar sont :

Pour l'exemple 1 :  $x^0 = (2.212543, 1.559423)^t$ .

Pour l'exemple 2 :  $x^0 = (0.013566, 0.141865, 0.110026, 0.268386, 0.110026, 0.233305)^t$ .

Pour l'exemple 3 :

$x^0 = (263.115358, 141.578025, 58.537897, 16.249702, 2.046456, 21.550602, 15.528872, 17.532554, 6.575938, 22.583593)^t$ .

Les solutions optimales sont :

Pour l'exemple 1 :  $x^* = (2.999999, 2)^t$ .

Pour l'exemple 2 :  $x^* = (0, 0.5, 0.000001, 0.499998, 0.000002, 0.000003)^t$ .

Pour l'exemple 3 :  $x^* = (841.112659, 0.000002, 0, 0, 0, 124.608341, 0, 0, 0, 0)^t$ .

Afin de réduire le nombre d'itérations et le temps de calcul, on a introduit une procédure pratique moins coûteuse que la recherche linéaire pour calculer le pas de déplacement. Si  $(x, y)$  est le point courant, alors le nouvel itéré est donné par :  $\hat{x} = x + \alpha\Delta x, \hat{y} = y + \alpha\Delta y$ . Puisque la méthode qu'on utilise est réalisable, alors le point  $(\hat{x}, \hat{y})$  doit être réalisable ( c'est à dire : il reste dans l'intérieur de la région réalisable). On a :  $\hat{y} = y + \alpha\Delta y = Mx + q + \alpha M\Delta x = M(x + \alpha\Delta x) + q = M\hat{x} + q$

Reste à vérifier que  $(\hat{x}, \hat{y}) > 0$ , pour cela on pose les conditions suivantes :

$$\begin{cases} x + \alpha_x \Delta x > 0 \\ y + \alpha_y \Delta y > 0 \end{cases}$$

On prend :  $\alpha = \beta \min(\alpha_x, \alpha_y)$  tel que  $0 < \beta < 1$ .

Où :

$$\alpha_x = \begin{cases} \min -\frac{x_i}{\Delta x_i} & \text{avec } i \in I = \{i : \Delta x_i < 0\} \\ 1 & \text{si } \Delta x \geq 0 \end{cases}$$

$$\alpha_y = \begin{cases} \min -\frac{y_i}{\Delta y_i} & \text{avec } i \in J = \{i : \Delta y_i < 0\} \\ 1 & \text{si } \Delta y \geq 0 \end{cases}$$

On a obtenu les résultats suivants :

	Nombre d'itérations	Temps de calcul (s)
<b>Exemple 1</b>	<b>66</b>	<b>0.03</b>
<b>Exemple 2</b>	<b>1214</b>	<b>0.86</b>
<b>Exemple 3</b>	<b>9444</b>	<b>20.76</b>

Les solutions optimales obtenues sont :

Pour l'exemple 1 :  $x^* = (2.999999, 1.999999)^t$ .

Pour l'exemple 2 :  $x^* = (0, 0.5, 0.000001, 0.499998, 0, 0.000005)^t$ .

Pour l'exemple 3 :  $x^* = (841.112669, 0.000002, 0, 0, 0, 124.608341, 0.000001, 0, 0, 0)^t$ .

En plus de la procédure pratique qui assure la réalisabilité des itérés, on pose :  $\mu^k = \frac{\mu^{k-1}}{10}$   
 (On ne tient pas compte des valeurs théoriques, on prend un choix libre).

Les solutions optimales obtenues sont :

Pour l'exemple 1 :  $x^* = (2.999999, 2)^t$ .

Pour l'exemple 2 :  $x^* = (0, 0.5, 0.000001, 0.499999, 0.000002, 0.000001)^t$ .

Pour l'exemple 3 :  $x^* = (841.112672, 0, 0, 0, 0, 124.608343, 0, 0, 0, 0)^t$ .

Les résultats obtenus sont présentés dans le tableau suivant :

	Nombre d'itérations	Temps de calcul (s)
<b>Exemple 1</b>	<b>7</b>	<b>00</b>
<b>Exemple 2</b>	<b>7</b>	<b>00</b>
<b>Exemple 3</b>	<b>11</b>	<b>0.06</b>

Les tests numériques présentés ici témoignent de l'importance des aménagements que nous avons apportés aux algorithmes de trajectoire centrale. En effet, le problème d'initialisation est réglé avec une réduction considérable du coût de l'itération ( temps de calcul et nombre d'itérations).

Dans ce qui suit, on présente des tests numériques de la dernière version de l'algorithme pour plusieurs exemples :

### Exemple 1

Soit le problème complémentaire linéaire monotone suivant :

$$(PCL) \left\{ \begin{array}{l} \text{Trouver } x \in \mathbb{R}^n \text{ tel que :} \\ y = Mx + q \\ x^t y = 0 \\ (x, y) \geq 0 \end{array} \right.$$

où :

$$M = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, q = \begin{pmatrix} -4 \\ -5 \\ -1 \end{pmatrix}$$

La solution est :  $x^* = (1, 2, 0)^t$ .

On considère les programmes quadratiques convexes suivants :

### Exemple 2

$$\begin{cases} \min x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_1 - 6x_2 \\ x_1 + x_2 \leq 2 \\ -x_1 + 2x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{cases}$$

La solution optimale est :  $x^* = (0.8, 1.2)^t$ .

### Exemple 3

$$\begin{cases} \min 4x_1^2 + 4x_2^2 + 6x_1x_2 - 2x_1 + 3x_2 \\ x_1 + 2x_2 \leq 6 \\ -x_1 + 2x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{cases}$$

La solution optimale est :  $x^* = (0.25, 0)^t$ .

Considérons les programmes linéaires suivants :

### Exemple 4

$$\begin{cases} \min x_1 + x_2 \\ 2x_1 + x_2 \leq 2 \\ -2x_1 + 4x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

La solution optimale est :  $x^* = (0, 0)^t$ .

**Exemple 5**

$$\left\{ \begin{array}{l} \min -4x_1 - 5x_2 \\ 2x_1 + x_2 \leq 8 \\ x_1 + 2x_2 \leq 7 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{array} \right.$$

La solution optimale est :  $x^* = (3, 2)^t$ .

**Exemple 6**

$$\left\{ \begin{array}{l} \min -x_1 - x_2 - x_3 \\ -x_1 + 2x_2 \leq 4 \\ x_1 - 3x_2 - x_3 \leq -3 \\ x_1 + x_2 \leq 9 \\ -x_1 - \frac{1}{2}x_2 + x_3 \leq -2 \\ -2x_2 \leq -5 \\ x_1, x_2, x_3 \geq 0 \end{array} \right.$$

La solution optimale est :  $x^* = (6.5, 2.5, 5.75)^t$ .

**Exemple 7**

$$\left\{ \begin{array}{l} \min 3x_1 - x_2 + x_3 \\ 2x_1 + x_2 - x_4 \leq 0 \\ x_3 + x_5 - x_6 \leq 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{array} \right.$$

La solution optimale est :  $x^* = (0, 0.5, 0, 0.5, 0, 0)^t$ .

### Exemple 8

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

où la matrice  $A$ , les vecteurs  $b$  et  $c$  sont donnés par :

$$A = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0.8 & 1 & 0 & 0 & 0 \\ 0.32 & 0.8 & 1 & 0 & 0 \\ 1.128 & 0.32 & 0.8 & 1 & 0 \\ 0.0512 & 0.128 & 0.32 & 0.8 & 1 \end{bmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, c = \begin{pmatrix} -0.0256 \\ -0.064 \\ -0.16 \\ -0.4 \\ -1 \end{pmatrix}$$

La solution optimale est :  $x^* = (0, 0, 0, 0, 1)^t$ .

### Exemple 9

$$A = \begin{bmatrix} 2 & -6 & 2 & 7 & 3 & 8 \\ -3 & -1 & 4 & -3 & 1 & 2 \\ 8 & -3 & 5 & -2 & 0 & 2 \\ 4 & 0 & 8 & 7 & -1 & 3 \\ 5 & 2 & -3 & 6 & -2 & -1 \end{bmatrix}, b = \begin{pmatrix} 1 \\ -2 \\ 4 \\ 1 \\ 5 \end{pmatrix}, c = \begin{pmatrix} -18 \\ 7 \\ -12 \\ -5 \\ 0 \\ -8 \end{pmatrix}$$

La solution optimale est :  $x^* = (2, 4, 0, 0, 7, 0)^t$ .

### Exemple 10

$$A = \begin{bmatrix} 1 & 0 & -4 & 3 & 1 & 1 \\ 5 & 3 & 1 & 0 & -1 & 3 \\ 4 & 5 & -3 & 3 & -4 & 1 \\ 0 & -1 & 0 & 2 & 1 & -5 \\ -2 & 1 & 1 & 1 & 2 & 2 \\ 2 & -3 & 2 & -1 & 4 & 5 \end{bmatrix}, b = \begin{pmatrix} 1 \\ 4 \\ 4 \\ 5 \\ 7 \\ 5 \end{pmatrix}, c = \begin{pmatrix} -4 \\ -5 \\ -1 \\ -3 \\ 5 \\ -8 \end{pmatrix}$$

La solution optimale est :  $x^* = (0, 0, 2.5, 3.5, 0, 0.5)^t$ .

### Exemple 11

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 5 & 4 & 3 & 2 & 1 \\ 6 & 7 & 8 & 9 & 10 & 5 & 2 & 8 & 3 & 1 \\ 11 & 12 & 13 & 14 & 15 & 6 & 7 & 80 & 90 & 10 \\ 1 & 10 & 20 & 30 & 40 & 50 & 60 & 80 & 90 & 10 \\ 3 & 9 & 27 & 60 & 45 & 60 & 75 & 8 & 9 & 46 \end{bmatrix}, b = \begin{bmatrix} 10000 \\ 10000 \\ 10000 \\ 10000 \\ 10000 \end{bmatrix}$$

$c_i = -1, i = 1, \dots, 10$ .

La solution optimale est :  $x^* = (841.121495, 0, 0, 0, 0, 124.610592, 0, 0, 0, 0)^t$ .

**Exemple 12**

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 5 & 4 & 3 & 2 & 1 \\ 6 & 7 & 8 & 9 & 10 & 5 & 2 & 8 & 3 & 1 \\ 11 & 12 & 13 & 14 & 15 & 6 & 7 & 80 & 90 & 10 \\ 1 & 10 & 20 & 30 & 40 & 50 & 60 & 80 & 90 & 10 \\ 3 & 9 & 27 & 60 & 45 & 60 & 75 & 8 & 9 & 46 \\ 90 & 100 & 100 & 20 & 30 & 1000 & 900 & 25 & 1 & 1 \\ 3 & 30 & 300 & 2 & 20 & 200 & 1 & 10 & 100 & 150 \\ 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 \\ 1 & 11 & 111 & 2 & 22 & 222 & 3 & 33 & 333 & 4 \\ 7 & 70 & 8 & 80 & 9 & 90 & 10 & 100 & 15 & 155 \end{bmatrix}$$

$b_i = 10000, c_i = -1$  pour  $i = 1, \dots, n$ .

La solution optimale est :

$$x^* = (50.147133, 0, 0, 18.009937, 155.909303, 0, 0, 15.979992, 17.503173, 31.898538)^t.$$

**Exemple 13**

$$A = \begin{bmatrix} 518 & 31 & 39 & 59 & 97 & 49 & 22 & 86 & 16 & 23 \\ 96 & 64 & 14 & 11 & 36 & 12 & 49 & 84 & 55 & 753 \\ 50 & 28 & 14 & 31 & 59 & 72 & 99 & 40 & 51 & 393 \\ 76 & 165 & 22 & 20 & 22 & 43 & 61 & 80 & 27 & 393 \\ 89 & 80 & 78 & 46 & 75 & 31 & 954 & 69 & 33 & 733 \\ 34 & 49 & 85 & 69 & 49 & 75 & 328 & 40 & 16 & 293 \\ 30 & 14 & 12 & 46 & 68 & 96 & 62 & 188 & 49 & 823 \\ 45 & 30 & 74 & 73 & 70 & 61 & 31 & 36 & 35 & 833 \\ 88 & 74 & 51 & 36 & 60 & 32 & 52 & 20 & 51 & 233 \\ 73 & 39 & 71 & 74 & 474 & 61 & 28 & 12 & 20 & 133 \end{bmatrix}$$

$b_i = 10000, c_i = -1$  pour  $i = 1, \dots, n$ .

La solution optimale est :

$$x^* = (0, 0, 7.555147, 58.860535, 0, 0, 0, 0, 146.974477, 0)^t.$$

Les résultats obtenus sont présentés dans le tableau suivant :

	taille	nombre d'itérations	temps de calcul (s)
<b>Exemple 1</b>	<b>3</b>	<b>6</b>	<b>00</b>
<b>Exemple 2</b>	<b>4</b>	<b>7</b>	<b>00</b>
<b>Exemple 3</b>	<b>4</b>	<b>6</b>	<b>00</b>
<b>Exemple 4</b>	<b>4</b>	<b>7</b>	<b>00</b>
<b>Exemple 5</b>	<b>5</b>	<b>7</b>	<b>00</b>
<b>Exemple 6</b>	<b>8</b>	<b>7</b>	<b>00</b>
<b>Exemple 7</b>	<b>9</b>	<b>7</b>	<b>00</b>
<b>Exemple 8</b>	<b>10</b>	<b>7</b>	<b>00</b>
<b>Exemple 9</b>	<b>11</b>	<b>9</b>	<b>00</b>
<b>Exemple 10</b>	<b>12</b>	<b>10</b>	<b>00</b>
<b>Exemple 11</b>	<b>15</b>	<b>11</b>	<b>0.06</b>
<b>Exemple 12</b>	<b>20</b>	<b>13</b>	<b>0.11</b>
<b>Exemple 13</b>	<b>20</b>	<b>15</b>	<b>0.11</b>

## Méthode de trajectoire centrale non réalisable

Dans les méthodes de trajectoire centrale réalisables, on a supposé que le point initial  $(x^0, y^0)$  est voisin de la trajectoire centrale, c'est à dire, il vérifie la contrainte  $y^0 = Mx^0 + q$ , la condition  $(x^0, y^0) > 0$  et l'inégalité  $\|XYe - \mu e\| \leq \mu\theta$ . Le calcul de ce point est fait par l'introduction des poids dans la fonction objectif du problème perturbé. Une autre alternative qui répond aux questions liées à l'initialisation est de démarrer l'algorithme d'un point initial qui vérifie seulement la condition  $(x^0, y^0) > 0$  (non nécessairement réalisable). En général, la suite générée par cet algorithme n'est pas réalisable d'où le nom : méthode de trajectoire centrale non réalisable.

### Description algorithmique

Soit  $(x^k, y^k) > 0$  le point obtenu à l'itération  $k$ . On applique la méthode de Newton pour déterminer la direction de déplacement  $(\Delta x^k, \Delta y^k)$  qui est solution du système linéaire suivant :

$$\begin{cases} M\Delta x^k - \Delta y^k = y^k - Mx^k - q \\ Y^k \Delta x^k + X^k \Delta y^k = \mu^k e - X^k Y^k e \end{cases}$$

D'où :

$$(*) \begin{cases} (M + (X^k)^{-1}Y^k) \Delta x^k = -q - Mx^k + \mu^k (X^k)^{-1}e \\ \Delta y^k = M\Delta x^k + Mx^k + q - y^k \end{cases}$$

Le nouvel itéré est donné par :  $(x^{k+1}, y^{k+1}) = (x^k, y^k) + \alpha(\Delta x^k, \Delta y^k)$  où  $\alpha > 0$  est le pas de déplacement.

Pour réaliser la réalisabilité des itérés et l'optimalité, on introduit une fonction dite de mérite définie par :

$$\phi(x, y) = \|XYe\|^2 + \|Mx + q - y\|^2$$

L'idée est de faire tendre la fonction  $\phi$  vers 0. Le pas de déplacement  $\alpha > 0$  est choisi de telle manière que l'itéré  $(x^{k+1}, y^{k+1})$  reste strictement positif ( $(x^{k+1}, y^{k+1}) > 0$ ) et que  $\phi$

décroit à chaque itération ( $\phi(x^{k+1}, y^{k+1}) < \phi(x^k, y^k)$ ).

L'algorithme se présente comme suit :

### Algorithme

#### Début algorithme

**Données :** Soient :  $\varepsilon > 0$  un paramètre de précision et  $\sigma_k \in (0, 1)$ .

**Initialisation :** soit :  $(x^0, y^0) > 0$

**Itération :** pour  $k = 0, 1, 2, \dots$  calculer  $\phi(x^k, y^k) = \phi^k$ .

1) **Si**  $\phi^k \leq \varepsilon$  **stop :**  $(x^k, y^k)$  est une solution approchée de (PCL).

2) Calculer  $\mu^k = \sigma_k \frac{(x^k)^t y^k}{n}$

3) Déterminer la direction de déplacement  $(\Delta x^k, \Delta y^k)$  solution du système (\*).

4) calculer le pas de déplacement  $\alpha^k > 0$  tel que :

$$\begin{cases} (x^{k+1}, y^{k+1}) > 0 \\ \phi^{k+1} < \phi^k \end{cases}$$

5) Poser :  $(x^{k+1}, y^{k+1}) = (x^k, y^k) + \alpha^k (\Delta x^k, \Delta y^k)$ .

6) Soit  $k = k + 1$  et aller à (1).

**Fin algorithme.**

#### Résultats de convergence

Pour démontrer la convergence de cet algorithme, on a besoin des lemmes suivants :

#### Lemme 1 [21]

Soit  $\{(x^k, y^k)\}$  une suite générée par l'algorithme, alors :

$$Mx^{k+1} + q - y^{k+1} = (1 - \alpha_k)(Mx^k + q - y^k) = \nu_{k+1}(Mx^0 + q - y^0)$$

$$\nu_{k+1} = (1 - \alpha_k)\nu_k = \prod_{j=0}^k (1 - \alpha_j) > 0 \text{ et } \nu_0 = 1. \quad \square$$

Le lemme suivant montre une réduction de la fonction de mérite à chaque itération.

**Lemme 2 [21]**

Soit  $\{(x^k, y^k)\}$  une suite générée par l'algorithme, alors :

$$\phi(x^{k+1}, y^{k+1}) \leq [1 - \alpha_k \beta (1 - \sigma_k)] \phi(x^k, y^k), \text{ où } \beta \in (0, \frac{1}{2}].$$

Si  $0 \leq \alpha_k \leq 1$ , alors la suite  $\{\phi^k\}$  converge au moins linéairement.  $\square$

Le théorème suivant montre que la suite  $\{\phi^k\}$  converge vers zéro.

**Théorème 9 [21]**

Soit  $\{(x^k, y^k)\}$  une suite générée par l'algorithme, alors pour tout  $\varepsilon > 0$  et  $\sigma_k \in (0, 1)$ ,  $\exists k^* :$

$$\phi(x^k, y^k) \leq \varepsilon, \forall k > k^*. \square$$

## Aspects numériques

Dans la mise en œuvre de l'algorithme de trajectoire centrale non réalisable, on démarre d'un point  $(x^0, y^0) > 0$ . pour calculer le pas de déplacement qui garantie les conditions suivantes :

$$\begin{cases} (x^{k+1}, y^{k+1}) > 0 \\ \phi^{k+1} < \phi^k \end{cases}$$

tel que :  $(x^{k+1}, y^{k+1}) = (x^k, y^k) + \alpha(\Delta x^k, \Delta y^k)$  où  $\alpha > 0$ , on propose la procédure suivante :

On prend :  $\alpha = \min(1, \alpha_x, \alpha_y)$ , où :

$$\alpha_x = \begin{cases} \min -\frac{x_i}{\Delta x_i} & \text{avec } i \in I = \{i : \Delta x_i < 0\} \\ 1 & \text{si } \Delta x \geq 0 \end{cases}$$

$$\alpha_y = \begin{cases} \min -\frac{y_i}{\Delta y_i} & \text{avec } i \in J = \{i : \Delta y_i < 0\} \\ 1 & \text{si } \Delta y \geq 0 \end{cases}$$

Nous présentons les résultats obtenus dans le tableau suivant :

	nombre d'itérations	temps de calcul (s)
<b>Exemple 1</b>	<b>6</b>	<b>0.01</b>
<b>Exemple 2</b>	<b>7</b>	<b>0.01</b>
<b>Exemple 3</b>	<b>7</b>	<b>0.01</b>
<b>Exemple 4</b>	<b>6</b>	<b>0.01</b>
<b>Exemple 5</b>	<b>8</b>	<b>0.01</b>
<b>Exemple 6</b>	<b>11</b>	<b>0.05</b>
<b>Exemple 7</b>	<b>7</b>	<b>0.01</b>
<b>Exemple 8</b>	<b>9</b>	<b>0.01</b>
<b>Exemple 9</b>	<b>14</b>	<b>0.01</b>
<b>Exemple 10</b>	<b>10</b>	<b>0.01</b>
<b>Exemple 11</b>	<b>35</b>	<b>0.11</b>
<b>Exemple 12</b>	<b>33</b>	<b>0.17</b>
<b>Exemple 13</b>	<b>53</b>	<b>0.27</b>

## Commentaires

On peut généraliser les techniques utilisées dans les méthodes de trajectoire centrale pour des classes de problèmes non linéaires. Cependant, cette généralisation doit s'effectuer progressivement. Dans le chapitre suivant, nous allons développer une variante de la méthode de trajectoire centrale pour la minimisation d'une fonction convexe sous contraintes linéaires.

## Chapter 2

Minimisation d'une fonction convexe  
sous contraintes linéaires par une  
méthode de trajectoire centrale avec  
poids

Nous proposons une variante primale-duale de trajectoire centrale pour minimiser une fonction convexe sous contraintes linéaires. C'est une approche barrière logarithmique avec poids qui offre une flexibilité dans le choix du point initial tout en conservant les propriétés théoriques.

On traite alors les conditions de KKT qui sont nécessaires et suffisantes par la méthode de Newton pour obtenir le nouvel itéré. L'algorithme proposé peut démarrer de n'importe quel point strictement réalisable sans supposer qu'il appartient au voisinage de la trajectoire centrale ( chemin central ).

## 2.1 Description de la méthode classique

On considère le problème d'optimisation suivant :

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

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , est une fonction non linéaire, convexe et de classe  $C^2$ ,  $A$  est une matrice de type  $(m, n)$  et de plein rang ( $rgA = m < n$ ) et  $b \in \mathbb{R}^m$ .

On note par :

$W = \{x \in \mathbb{R}_+^n : Ax = b\}$  l'ensemble des solutions réalisables de  $(PNL)$ .

$W_{int} = \{x \in \mathbb{R}_{++}^n : Ax = b\}$  l'ensemble des solutions strictement réalisables de  $(PNL)$  qui est supposé non vide.

On associe à  $(PNL)$  le problème perturbé suivant :

$$(PNL)_\mu \begin{cases} \min f_\mu(x) = \left( f(x) - \mu \sum_{i=1}^n \ln x_i \right) \\ Ax = b \\ x > 0 \end{cases}$$

$\mu > 0$  est le paramètre barrière.

$f_\mu$  est une fonction strictement convexe, il suffit de voir que :  $\forall x > 0$  on a :

$$\nabla f_\mu(x) = \nabla f(x) - \mu X^{-1}e$$

$\nabla^2 f_\mu(x) = \nabla^2 f(x) + \mu X^{-2}$  est une matrice définie positive ( puisque  $f$  est convexe ).

$(PNL)_\mu$  est un problème strictement convexe (les contraintes sont affines), donc les conditions d'optimalité de KKT sont nécessaires et suffisantes, elles s'écrivent comme suit :

$$\left\{ \begin{array}{l} \nabla f(x) - \mu X^{-1}e + A^t y = 0 \\ Ax = b \\ x > 0 \end{array} \right.$$

qui est équivalent au système suivant en posant  $s = \mu X^{-1}e$  :

$$(*) \left\{ \begin{array}{l} A^t y + s = \nabla f(x) \\ Ax = b \\ Xs - \mu e = 0 \\ (x, s) > 0 \end{array} \right.$$

Pour chaque valeur de  $\mu$ , on résout le système (\*) par la méthode de Newton, on obtient ainsi une suite de solutions paramétrisées qui converge vers la solution du problème initial lorsque  $\mu$  tend vers 0.

On définit l'ensemble :

$$T = \{(x, y, s) \in W_{int} \times \mathbb{R}^m \times \mathbb{R}_{++}^n : A^t y + s = \nabla f(x)\}$$

La trajectoire centrale est l'ensemble de toutes les solutions du système(\*), elle est notée par :

$$TC = \{(x, y, s) \in T : XSe - \mu e = 0\}$$

Puisqu'on a une équation non linéaire dans le système (\*), il est généralement impossible

d'obtenir une solution exacte de  $(PNL)_\mu$ . Nous allons donc nous contenter d'une solution approchée ( la méthode de Newton est privilégiée dans ce cas ) dont la qualité est jugée satisfaisante lorsqu'elle se trouve dans le voisinage de la trajectoire centrale. Ce voisinage correspond aux conditions de KKT où l'on remplace l'équation non linéaire par la contrainte :  $\|XSe - \mu e\| \leq \theta\mu$  avec  $0 < \theta < 1$ . Un point est dit voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$T(\theta) = \{(x, y, s) \in T : \|XSe - \mu e\| \leq \theta\mu, 0 < \theta < 1\}$$

Les méthodes primales duales déterminent les solutions du système (\*) en appliquant la méthode de Newton comme suit : on définit la fonction non linéaire :

$$F_\mu : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$$

$$F_\mu(x, y, s) = \begin{pmatrix} A^t y + s - \nabla f(x) \\ Ax - b \\ XSe - \mu e \end{pmatrix}$$

où  $X = \text{diag}(x_1, \dots, x_n)$ ,  $S = \text{diag}(s_1, \dots, s_n)$ .

Pour  $(x, y, s) \in T$ , on obtient le système :

$$\begin{cases} A\Delta x = 0 \\ \nabla^2 f(x)\Delta x - A^t\Delta y - \Delta s = 0 \\ X\Delta s + S\Delta x = -Xs + \mu e \end{cases}$$

dont la solution est  $(\Delta x, \Delta y, \Delta s)$  et le nouvel itéré est :  $(\hat{x}, \hat{y}, \hat{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$ .

L'algorithme correspondant se présente comme suit :

**Algorithme**

**Début algorithme**

**Initialisation :**  $\varepsilon > 0$  est un paramètre de précision et  $\theta, \beta \in (0, 1)$  sont des constantes données.

Soit  $(x^0, y^0, s^0) \in T(\theta)$ .

**Itération :** Pour  $k = 0, 1, 2, \dots$

- Calculer :  $\mu_k = \frac{(x^k)^t s^k}{n}$  et
- Résoudre le système :

$$\begin{bmatrix} S^k & 0 & X^k \\ \nabla^2 f(x^k) & -A^t & -I \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -X^k S^k e + \beta \mu_k e \\ 0 \\ 0 \end{bmatrix}$$

pour obtenir:  $(\Delta x^k, \Delta y^k, \Delta s^k)$ .

- Prendre :  $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$  et soit  $k = k + 1$ .

**jusqu'à**  $(x^k)^t s^k \leq \varepsilon$

**Fin algorithme.**

Si le point initial calculé n'est pas voisin de la trajectoire centrale, rien ne garantie la convergence de cet algorithme même si ce point est strictement réalisable. Cette difficulté peut être surmontée en introduisant des poids dans la fonction barrière.

## 2.2 Méthode de trajectoire centrale avec poids

### Description

On considère de nouveau le problème non linéaire :

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

On associe à (PNL) le problème perturbé défini comme suit :

$$(PNL)_{\mu r} \begin{cases} \min f_{\mu r}(x) = \left( f(x) - \mu \sum_{i=1}^n r_i \ln x_i \right) \\ Ax = b \\ x > 0 \end{cases}$$

$\mu > 0, r = (r_1, r_2, \dots, r_n)^t \in \mathbb{R}_+^n$  est le vecteur des poids associé à la fonction barrière.

$f_{\mu r}(x)$  est strictement convexe et les contraintes sont affines, alors les conditions de KKT correspondantes sont nécessaires et suffisantes et s'écrivent :

$$\begin{cases} \nabla f(x) - \mu X^{-1}r + A^t y = 0 \\ Ax = b \\ x > 0 \end{cases}$$

ou encore sous la forme :

$$(**) \begin{cases} A^t y + s = \nabla f(x) \\ Ax = b \\ Xs - \mu r = 0 \\ (x, s) > 0 \end{cases}$$

avec :  $s = \mu X^{-1}r$  et  $(-y) = y \in \mathbb{R}^m$ .

Pour  $r = e$  dans le système (\*\*), on trouve le système (\*), c'est à dire la méthode primale duale de trajectoire centrale sans poids.

On applique alors la méthode de Newton pour  $(x, y, s) \in T$ , on obtient le système :

$$\begin{cases} A\Delta x = 0 \\ \nabla^2 f(x)\Delta x - A^t\Delta y - \Delta s = 0 \\ X\Delta s + S\Delta x = -Xs + \mu r \end{cases}$$

dont la solution est  $(\Delta x, \Delta y, \Delta s)$ . Le nouvel itéré est :  $(\hat{x}, \hat{y}, \hat{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$ .

Le point  $(x_\mu, y_\mu, s_\mu)$  qui vérifie le système (\*\*\*) est un point de la trajectoire centrale. Ce dernier est voisin de la trajectoire centrale s'il appartient à l'ensemble :

$$T_r(\theta) = \{(x, y, s) \in T / \|XSe - \mu r\| \leq \theta\mu, 0 < \theta < 1\}$$

Nous proposons maintenant notre algorithme de trajectoire centrale avec poids pour la résolution d'un programme non linéaire.

### Algorithme

#### Début algorithme

**Initialisation :**  $\varepsilon > 0$  est un paramètre de précision et  $\beta \in (0, 1)$ .

Soit :  $(x^0, y^0, s^0) \in T$ , on pose:  $\mu = \frac{\|X^0 S^0 e\|}{\sqrt{n}}$  et  $r = \frac{X^0 S^0 e}{\mu}$ .

**Itération :** pour  $k = 0, 1, 2, \dots$

**Si :**  $(x^k)^t R s^k \leq \varepsilon$ , Stop.

**Sinon :**

- Calculer :  $\mu^k = \frac{(x^k)^t R s^k}{n}$ .

- Résoudre le système :

$$\begin{bmatrix} S^k & 0 & X^k \\ \nabla^2 f(x^k) & -A^t & -I \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -X^k S^k e + \beta \mu^k r \\ 0 \\ 0 \end{bmatrix}$$

pour obtenir :  $(\Delta x^k, \Delta y^k, \Delta s^k)$ .

- Soit :  $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$ .
- Poser  $k = k + 1$ .

**Fin algorithme.**

## Calcul de la direction de déplacement

Le calcul le plus coûteux de chaque itération de l'algorithme est la résolution du système linéaire suivant :

$$(1) \begin{cases} A\Delta x = 0 & (a) \\ \nabla^2 f(x)\Delta x - A^t\Delta y - \Delta s = 0 & (b) \\ S\Delta x + X\Delta s = -Xs + \hat{\mu}r & (c) \end{cases}$$

On propose la procédure suivante pour la résolution :

De l'équation (1.b) on a :  $\Delta s = \nabla^2 f(x)\Delta x - A^t\Delta y$ .

On remplace dans (1.c) :  $(X\nabla^2 f(x) + S)\Delta x - XA^t\Delta y = -Xs + \hat{\mu}r$ .

On pose :  $H = \nabla^2 f(x) + X^{-1}S$ , on obtient le système :

$$\begin{cases} A\Delta x = 0 \\ H\Delta x - A^t\Delta y = X^{-1}(-Xs + \hat{\mu}r) \end{cases}$$

$\Delta x = H^{-1}A^t\Delta y + H^{-1}[-Se + \hat{\mu}X^{-1}r]$ , on a :  $(AH^{-1}A^t)\Delta y = AH^{-1}(Se - \hat{\mu}X^{-1}r)$ .

La solution est donnée par :

$$(2) \begin{cases} (AH^{-1}A^t)\Delta y = AH^{-1}(Se - \hat{\mu}X^{-1}r) \\ \Delta x = H^{-1}A^t\Delta y - H^{-1}(Se - \hat{\mu}X^{-1}r) \\ \Delta s = \nabla^2 f(x)\Delta x - A^t\Delta y \end{cases}$$

## Convergence

Soit  $(x, y, s)$  un point se trouvant dans le voisinage du chemin central c'est à dire :

$$(x, y, s) \in T \text{ et } \|XSe - \mu r\| = \left[ \sum_{i=1}^n (x_i s_i - \mu r_i)^2 \right]^{\frac{1}{2}} \leq \theta \mu \text{ où } \mu = \frac{x^t R s}{n} \text{ et } \theta \in (0, 1).$$

On pose :  $\hat{\mu} = \beta \mu, \beta \in (0, 1)$ .

Soit  $(\hat{x}, \hat{y}, \hat{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$  où  $(\Delta x, \Delta y, \Delta s)$  est la solution du système suivant

:

$$(3) \begin{cases} A\Delta x = 0 & (a) \\ A^t \Delta y + \Delta s = \nabla^2 f(x) \Delta x & (b) \\ X\Delta s + S\Delta x = -Xs + \hat{\mu} r & (c) \end{cases}$$

On pose :  $D = (XS^{-1})^{\frac{1}{2}}, \Delta X = \text{diag}(\Delta x), \hat{X} = \text{diag}(\hat{x})$  et  $\hat{S} = \text{diag}(\hat{s})$ .

Pour démontrer la convergence de notre algorithme, on a besoin des quatres lemmes suivants :

### Lemme 1

Soit  $\eta = \min_i r_i$  pour  $i = 1, \dots, n$ , et  $(x, y, s) \in T_r(\theta)$ , alors:

$$1) \max_i \{x_i s_i\} \leq (\eta + \theta) \mu$$

$$2) \min_i \{x_i s_i\} \geq (\eta - \theta) \mu .$$

### Preuve

1)  $(x, y, s)$  vérifie la condition :  $\|XSe - \mu r\| \leq \theta \mu$ , donc :

$$|x_i s_i - \mu r_i| \leq \left[ \sum_{i=1}^n (x_i s_i - \mu r_i)^2 \right]^{\frac{1}{2}} = \|XSe - \mu r\| \leq \theta \mu$$

$$\left( x_i^2 \leq \sum_{i=1}^n x_i^2 \Rightarrow |x_i| \leq \left[ \sum_{i=1}^n x_i^2 \right]^{\frac{1}{2}} = \|x\| \right)$$

$$\Rightarrow (r_i - \theta) \mu \leq x_i s_i \leq (r_i + \theta) \mu$$

$$\Rightarrow (\eta - \theta) \mu \leq x_i s_i \leq (\eta + \theta) \mu$$

$$x_i s_i \leq (\eta + \theta) \mu \Rightarrow \max_i \{x_i s_i\} \leq (\eta + \theta) \mu$$

2) De la même façon on a :  $x_i s_i \geq (\eta - \theta)\mu \Rightarrow \min_i \{x_i s_i\} \geq (\eta - \theta)\mu$ .  $\square$

**Lemme 2**

Pour  $\eta > \theta$ , on a :

$$\left\| (XS)^{-\frac{1}{2}} \right\|_2 \leq \frac{1}{\sqrt{(\eta - \theta)\mu}}$$

**Preuve**

Du lemme 1, on a :

$$\begin{aligned} x_i s_i &\geq (\eta - \theta)\mu \\ \Rightarrow \frac{1}{\sqrt{x_i s_i}} &\leq \frac{1}{\sqrt{(r_i - \theta)\mu}} \leq \frac{1}{\sqrt{(\eta - \theta)\mu}} \\ \Rightarrow \left\| (XS)^{-\frac{1}{2}} \right\|_2 &= \max \left\{ \frac{1}{\sqrt{x_i s_i}} \right\} \leq \frac{1}{\sqrt{(\eta - \theta)\mu}}. \quad \square \end{aligned}$$

**Lemme 3**

Soit  $z = (XS)^{-\frac{1}{2}}(-Xs + \hat{\mu}r)$  alors :

$$\|z\| \leq \sqrt{\frac{(\theta^2 + \delta^2)\mu}{(\eta - \theta)}}$$

**Preuve**

$$\begin{aligned} \|z\|^2 &= \left\| (XS)^{-\frac{1}{2}}(\hat{\mu}r - XSe) \right\|^2 \\ &= \left\| (XS)^{-\frac{1}{2}}[(\mu r - XSe) + (\hat{\mu} - \mu)r] \right\|^2 \\ &\leq \left\| (XS)^{-\frac{1}{2}} \right\|_2^2 [\|\mu r - XSe\|^2 + \|(\hat{\mu} - \mu)r\|^2] \\ &\leq \frac{1}{(\eta - \theta)\mu} [\theta^2 \mu^2 + (\hat{\mu} - \mu)^2 n] \\ &\leq \frac{1}{(\eta - \theta)\mu} [\theta^2 \mu^2 + (1 - \beta)^2 \mu^2 n] \\ &\leq \frac{\mu}{(\eta - \theta)} [\theta^2 + (1 - \beta)^2 n] \end{aligned}$$

On pose :  $(1 - \beta)^2 n = \delta^2$  donc :

$$\|z\|^2 \leq \frac{\mu}{(\eta - \theta)} [\theta^2 + \delta^2] = \left( \frac{\theta^2 + \delta^2}{\eta - \theta} \right) \mu$$

$$\|z\| \leq \sqrt{\frac{(\theta^2 + \delta^2)}{(\eta - \theta)}} \mu. \quad \square$$

**Lemme 4**

On a :

$$1) \left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| \leq \frac{(\theta^2 + \delta^2)}{2(\eta - \theta)} \mu$$

$$2) \left| \frac{\widehat{x}^t R \widehat{s}}{n} - \widehat{\mu} \right| \leq \frac{(\theta^2 + \delta^2)}{2\sqrt{n}(\eta - \theta)} \mu$$

**Preuve**

1) De l'équation (3.c), On a :  $D\Delta s + D^{-1}\Delta x = z$  c'est à dire :

$$\sqrt{\frac{x_i}{s_i}} \Delta s_i + \sqrt{\frac{s_i}{x_i}} \Delta x_i = \frac{\widehat{\mu} r_i - x_i s_i}{\sqrt{x_i s_i}}$$

On pose :  $u_i = \sqrt{\frac{x_i}{s_i}} \Delta s_i$  et  $v_i = \sqrt{\frac{s_i}{x_i}} \Delta x_i$ , donc

$$\begin{cases} u = D\Delta s \\ v = D^{-1}\Delta x \\ u_i + v_i = z_i \end{cases}$$

De l'équation (3.c), On a :

$$\widehat{x}_i \widehat{s}_i - \widehat{\mu} r_i = (x_i + \Delta x_i)(s_i + \Delta s_i) - \widehat{\mu} r_i = x_i s_i - \widehat{\mu} r_i + (s_i \Delta x_i + x_i \Delta s_i) + \Delta x_i \Delta s_i$$

alors :  $\widehat{X} \widehat{S} e - \widehat{\mu} r = \Delta X \Delta s$

$$\left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| = \left[ \sum_{i=1}^n (\widehat{x}_i \widehat{s}_i - \widehat{\mu} r_i)^2 \right]^{\frac{1}{2}} = \left[ \sum_{i=1}^n (\Delta x_i \Delta s_i)^2 \right]^{\frac{1}{2}} = \left[ \sum_{i=1}^n (u_i v_i)^2 \right]^{\frac{1}{2}}$$

On a :

$$\sum_{i=1}^n (u_i v_i)^2 \leq \left( \sum_{i=1}^n u_i v_i \right)^2 \Rightarrow \left[ \sum_{i=1}^n (u_i v_i)^2 \right]^{\frac{1}{2}} \leq \sum_{i=1}^n u_i v_i$$

$$\left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| \leq \sum_{i=1}^n u_i v_i \leq \frac{1}{2} \sum_{i=1}^n (u_i + v_i)^2 = \frac{1}{2} \sum_{i=1}^n z_i^2 = \frac{1}{2} \|z\|^2 \leq \frac{\theta^2 + \delta^2}{2(\eta - \theta)} \mu$$

$$\left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| \leq \frac{(\theta^2 + \delta^2)}{2(\eta - \theta)} \mu$$

2)

$$\begin{aligned} \left| \frac{\widehat{x}^t R \widehat{s}}{n} - \widehat{\mu} \right| &= \left| \frac{\widehat{x}^t R \widehat{s}}{n} - \widehat{\mu} \frac{r^t r}{n} \right| = \frac{1}{n} \left| \widehat{x}^t R \widehat{s} - \widehat{\mu} r^t r \right| \\ &= \frac{1}{n} \left| \sum_{i=1}^n \widehat{x}_i r_i \widehat{s}_i - \widehat{\mu} r^t r \right| = \frac{1}{n} \left| r^t (\widehat{X} \widehat{S} e - \widehat{\mu} r) \right| \\ &\leq \frac{1}{n} \|r\| \left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| \leq \frac{(\theta^2 + \delta^2)}{2\sqrt{n}(\eta - \theta)} \mu. \quad \square \end{aligned}$$

### Remarque 1

Du lemme 4, on a :

$$\left| \frac{\widehat{x}^t R \widehat{s}}{n} - \widehat{\mu} \right| \leq \frac{(\theta^2 + \delta^2)}{2\sqrt{n}(\eta - \theta)} \mu$$

alors :

$$\widehat{\mu} - \frac{(\theta^2 + \delta^2)}{2\sqrt{n}(\eta - \theta)} \mu \leq \frac{\widehat{x}^t R \widehat{s}}{n} \leq \widehat{\mu} + \frac{(\theta^2 + \delta^2)}{2\sqrt{n}(\eta - \theta)} \mu$$

$$\left( 1 - \frac{(\theta - \delta)^2 + 2\delta\eta}{2\sqrt{n}(\eta - \theta)} \right) \mu \leq \frac{\widehat{x}^t R \widehat{s}}{n} \leq \left( 1 + \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} \right) \mu \quad (4)$$

**Choix des paramètres :** On choisit maintenant les paramètres  $\theta, \beta, \delta$  tels que les conditions (5) et (6) ci-dessous soient réalisées :

1) Pour rester dans le voisinage de la trajectoire centrale c'est à dire :

$$\left\| \widehat{X} \widehat{S} e - \widehat{\mu} r \right\| \leq \theta \beta \mu = \theta \widehat{\mu}$$

on pose la condition suivante :

$$\frac{\theta^2 + \delta^2}{2(\eta - \theta)} \leq \beta \theta \quad (5)$$

2) On veut aussi que :  $\widehat{x}^t R \widehat{s} \leq C(x^t R s)$  avec  $C < 1$ .

On a :

$$\frac{\widehat{x}^t R \widehat{s}}{n} \leq \left( 1 + \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} \right) \mu$$

Donc :

$$\left( 1 + \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} \right) < 1$$

Alors :

$$\frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} < 0 \quad (6)$$

On a les conditions suivantes :

$$\left\{ \begin{array}{l} \eta > \theta \\ \frac{\theta^2 + \delta^2}{2(\eta - \theta)} \leq \beta \theta, \delta > 0 \\ \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} < 0 \end{array} \right.$$

Puisque :  $(1 - \beta)^2 \eta = \delta^2$  alors  $\beta = 1 - \frac{\delta}{\sqrt{\eta}}$

$$\Leftrightarrow \begin{cases} \eta > \theta \\ \frac{\theta^2 + \delta^2}{2(\eta - \theta)} \leq \left(1 - \frac{\delta}{\sqrt{n}}\right) \theta \\ \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)} < 0 \end{cases} \quad (7)$$

La solution la plus simple du système (7) est de prendre :  $\theta = \delta$ , on obtient :

$$\begin{cases} \frac{\theta^2}{\eta - \theta} \leq \left(1 - \frac{\theta}{\sqrt{n}}\right) \theta \\ \frac{4\theta^2 - 2\theta\eta}{2(\eta - \theta)} < 0 \end{cases} \quad (8)$$

$$\Leftrightarrow \begin{cases} \theta^2 - (2\sqrt{n} + \eta)\theta + \eta\sqrt{n} \geq 0 \\ \theta < \frac{\eta}{2} \end{cases}$$

On peut vérifier que pour  $0 < \theta < \frac{\eta}{3}$ , les deux conditions du système (8) sont satisfaites. Maintenant, On veut calculer la valeur optimale de  $\theta$  dans le sens que le taux de diminution soit le maximum. Sous les conditions (8), on veut rendre le terme :

$1 + \frac{(\theta + \delta)^2 - 2\delta\eta}{2\sqrt{n}(\eta - \theta)}$  aussi petit que possible. On définit donc la fonction :

$$g(\theta) = \frac{2\theta^2 - \eta\theta}{\eta - \theta}$$

On a :

$$g'(\theta) = \frac{-2\theta^2 + 4\eta\theta - \eta^2}{(\eta - \theta)^2}$$

Le point critique unique de  $g$  dans l'intervalle  $(0, \eta)$  est  $\theta^* = \left(1 - \frac{\sqrt{2}}{2}\right) \eta$ .

De plus, on a :

$$g''(\theta) = \frac{2\eta^2}{(\eta - \theta)^3} > 0$$

Puisque  $g''(\theta^*) = \frac{4\sqrt{2}}{\eta} > 0$ , alors  $\theta^* = \left(1 - \frac{\sqrt{2}}{2}\right)\eta$  est un minimum global de  $g$  dans  $(0, \eta)$ .

On a donc le théorème suivant :

**Théorème 1**

Soient  $\theta = \delta = \left(1 - \frac{\sqrt{2}}{2}\right)\eta$  et  $\beta = \left(1 - \frac{\delta}{\sqrt{n}}\right)$ , supposons que  $(x, y, s) \in T_r(\theta)$  et

$\hat{\mu} = \beta\mu$  tel que  $\mu = \frac{x^t R s}{n}$  alors :

1)  $(\hat{x}, \hat{y}, \hat{s}) \in T_r(\theta)$  où  $(\hat{x}, \hat{y}, \hat{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$ .

2)  $\hat{x}^t R \hat{s} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) x^t R s$

**Preuve**

1) On a :  $A\hat{x} = A(x + \Delta x) = Ax + A\Delta x$

De l'équation (3.a) on a:  $A\Delta x = 0$  et  $x$  est réalisable, donc :

$$A\hat{x} = Ax = b$$

De l'équation (3.b) on a :

$$\begin{aligned} A^t \hat{y} + \hat{s} &= A^t(y + \Delta y) + s + \Delta s \\ &= A^t y + s + (A^t \Delta y + \Delta s) = \nabla f(x) + (A^t \Delta y + \Delta s) \\ &= \nabla f(x) + \nabla^2 f(x) \Delta x = \nabla f(x + \Delta x) = \nabla f(\hat{x}) \end{aligned}$$

De plus, sous les conditions précédentes, on peut démontrer que l'on reste dans le voisinage de la trajectoire centrale, c'est à dire :

$$\left\| \hat{X} \hat{S} e - \hat{\mu} r \right\| \leq \theta \hat{\mu}$$

Il reste à vérifier que :  $(\hat{x}, \hat{s}) > 0$ .

On a :

$$|\hat{x}_i \hat{s}_i - \hat{\mu} r_i| \leq \left\| \hat{X} \hat{S} e - \hat{\mu} r \right\| \leq \theta \hat{\mu}$$

$$-\theta \hat{\mu} \leq \hat{x}_i \hat{s}_i - \hat{\mu} r_i \leq \theta \hat{\mu} \Rightarrow (r_i - \theta) \hat{\mu} \leq \hat{x}_i \hat{s}_i \leq (r_i + \theta) \hat{\mu}$$

$$\hat{x}_i \hat{s}_i \geq (r_i - \theta) \hat{\mu} \geq (\eta - \theta) \hat{\mu} > 0 \text{ puisque } \eta > \theta$$

donc  $\hat{x}_i$  et  $\hat{s}_i$  sont de même signe. On suppose que  $\hat{x}_i < 0, \hat{s}_i < 0$ , alors :

$$\hat{x}_i = x_i + \Delta x_i \Rightarrow \Delta x_i = \hat{x}_i - x_i < 0$$

$$|\Delta x_i| = -\Delta x_i = x_i - \hat{x}_i > -\hat{x}_i = |-\hat{x}_i| \Rightarrow |\Delta x_i| > |-\hat{x}_i|$$

De même on a :  $|\Delta s_i| > |-\hat{s}_i|$

$$\text{Donc : } \Delta x_i \Delta s_i = (-|\Delta x_i|)(-|\Delta s_i|) = |\Delta x_i| |\Delta s_i|$$

$$\Delta x_i \Delta s_i = |\Delta x_i| |\Delta s_i| > |-\hat{x}_i| |-\hat{s}_i| = |\hat{x}_i \hat{s}_i| = \hat{x}_i \hat{s}_i \text{ ( puisque } \hat{x}_i \hat{s}_i > 0 \text{ )}.$$

$$\text{Donc : } \Delta x_i \Delta s_i > \hat{x}_i \hat{s}_i$$

En contradiction avec :  $\Delta x_i \Delta s_i = \hat{x}_i \hat{s}_i - \hat{\mu} r_i < \hat{x}_i \hat{s}_i$ , d'où  $\hat{x}_i > 0, \hat{s}_i > 0$ .

$$2) \text{ On a : } g(\theta^*) = -(\sqrt{2} - 1)^2 \eta < -\frac{\eta}{6}$$

$$C = 1 + \frac{1}{\sqrt{n}} g(\theta^*) < 1 - \frac{\eta}{6\sqrt{n}}. \quad \square$$

## Théorème 2

Soit  $\varepsilon > 0$  une précision donnée. Supposons que notre algorithme génère une suite des itérés  $(x^k, y^k, s^k)$  qui satisfait :  $(x^{k+1})^t R s^{k+1} \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) (x^k)^t R s^k$  pour  $\eta > 0$ , alors

il existe  $K$  avec  $K = O \left[ \sqrt{n} \ln \left( \frac{(x^0)^t R s^0}{\varepsilon} \right) \right]$  tel que :  $(x^k)^t R s^k \leq \varepsilon$  pour tout  $k \geq K$ .

Si de plus, le point initial  $(x^0, y^0, s^0)$  vérifie la condition :  $(x^0)^t R s^0 < 2^{2L}$  et  $\varepsilon = 2^{-2L}$ , alors l'algorithme s'arrête au bout de  $\left[ \left( \frac{24 \ln 2}{\eta} L \right) \sqrt{n} \right]$  itérations.

$L$  représente la taille du problème exprimée en nombre de bits requis pour représenter les données de (PNL).

## Preuve

Pour chaque  $k$  on a :

$$(x^k)^t R s^k \leq \left(1 - \frac{\eta}{6\sqrt{n}}\right) (x^{k-1})^t R s^{k-1}$$

$$\ln((x^k)^t R s^k) \leq \ln \left(1 - \frac{\eta}{6\sqrt{n}}\right) + \ln((x^{k-1})^t R s^{k-1})$$

On obtient :

$$\ln((x^k)^t R s^k) \leq k \ln \left(1 - \frac{\eta}{6\sqrt{n}}\right) + \ln((x^0)^t R s^0)$$

Puisque :  $\ln(1 + \beta) \leq \beta$  pour tout  $\beta > -1$ , alors :

$$\ln((x^k)^t R s^k) \leq k \left( -\frac{\eta}{6\sqrt{n}} \right) + \ln((x^0)^t R s^0)$$

Pour que :  $(x^k)^t R s^k \leq \varepsilon$ , on a :

$$k \left( -\frac{\eta}{6\sqrt{n}} \right) + \ln((x^0)^t R s^0) \leq \ln \varepsilon$$

$$k \left( -\frac{\eta}{6\sqrt{n}} \right) \leq \ln \varepsilon - \ln((x^0)^t R s^0) = \ln \left( \frac{\varepsilon}{(x^0)^t R s^0} \right)$$

$$k \geq K = \left\lceil \frac{6\sqrt{n}}{\eta} \ln \left( \frac{(x^0)^t R s^0}{\varepsilon} \right) \right\rceil$$

Si  $(x^0)^t R s^0 < 2^{2L}$  et  $\varepsilon = 2^{-2L}$  alors :  $k \geq \left( \frac{24 \ln 2}{\eta} L \right) \sqrt{n}$ .  $\square$

## Chapter 3

Une variante de type projectif pour  
la minimisation d'une fonction  
convexe sous contraintes linéaires

Nous proposons dans ce chapitre une méthode de points intérieurs de type projectif pour minimiser une fonction non linéaire sur un polyèdre convexe. C'est une extension de la méthode de Karmarkar pour la programmation linéaire que nous rappelons brièvement ici.

### 3.1 Méthode de Karmarkar

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 :

$$(PLS) \begin{cases} \min c^t x = z^* \\ Ax = 0 \\ x \in S_n \end{cases}$$

pour lequel on connaît à priori la valeur optimale  $z^* = 0$  et une solution réalisable, par exemple le point  $a = \frac{1}{n}e_n$  (centre du simplexe  $S_n$ ),  $e_n$  désigne le vecteur dont toutes les composantes sont égales à 1.

$A$  est une matrice de type  $(m, n)$  et de plein rang ( $rgA = m < n$ ) et  $S_n = \{x \in \mathbb{R}^n : x \geq 0, e_n^t x = 1\}$  est le simplexe de dimension  $(n - 1)$ .

**Remarque :**

1) Si la valeur optimale  $z^*$  est non nulle, alors l'égalité  $e_n^t x = 1$  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 :  $c^t x^* = z^* = z^* e_n^t x^* \Rightarrow (c - z^* e_n)^t x^* = \widehat{c}^t x^* = 0$ .

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

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

$$Ax = be_n^t x \Rightarrow (A - be_n^t)x = 0.$$

## 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 :

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

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 :

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

On obtient le programme linéaire équivalent suivant :

$$(2) \left\{ \begin{array}{l} \min c^t D_k y \\ A D_k y = 0 \\ e_n^t y = 1 \\ y \geq 0 \end{array} \right.$$

qui est bien de la forme (PLS).

On a besoin du lemme suivant pour simplifier la résolution du problème (2) ci-dessus :

**Lemme 1**

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

$$E = \left\{ y \in \mathbb{R}^n : \sum_{i=1}^n \frac{(y_i - y_i^0)^2}{(y_i^0)^2} \leq \beta^2, 0 < \beta < 1 \right\}$$

est dans l'intérieur de l'orthant positif de  $\mathbb{R}^n$ .

**Preuve**

Supposons le contraire, c'est à dire : il existe  $j \in \{1, \dots, n\}$  tel que:  $y_j \leq 0$  alors :

$$\sum_{i=1}^n \frac{(y_i - y_i^0)^2}{(y_i^0)^2} \geq \frac{(y_j - y_j^0)^2}{(y_j^0)^2} \geq 1 > \beta^2. \square$$

D'après le lemme 1, si on ajoute au problème (2) la contrainte :

$\{y \in \mathbb{R}^n / \|y - a\| \leq \alpha r\}$  où  $0 < \alpha < 1$  et  $r = \frac{1}{\sqrt{n(n-1)}}$ , alors la contrainte de positivité  $y \geq 0$  devient redondante. On obtient alors le problème :

$$(3) \left\{ \begin{array}{l} \min c^t D_k y \\ AD_k y = 0 \\ e_n^t y = 1 \\ \|y - a\|^2 \leq (\alpha r)^2 \end{array} \right.$$

C'est la minimisation d'une fonction linéaire sur une sphère, où la solution optimale est triviale conformément au théorème suivant :

**Théorème 1 [13]**

La solution optimale du problème (3) est donnée explicitement par :  $y^k = a - \alpha r d^k$

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

**Preuve**

Il suffit d'écrire les conditions nécessaires et suffisantes d'optimalité de KKT.  $\square$

A chaque itération, on revient à la variable initiale  $x$  en appliquant la transformation inverse  $T_k^{-1}$  et ainsi de suite jusqu'à ce que le test d'optimalité ( $c^t x \leq \varepsilon$ ) se réalise, où  $\varepsilon > 0$  est une précision donnée.

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

## Convergence de l'algorithme

On a le théorème suivant :

### Théorème 2 [13]

Le point  $y^k$  vérifie :

$$\frac{c^t D_k y^k}{c^t D_k a} \leq 1 - \frac{\alpha}{n-1}. \quad \square$$

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

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

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

## Lemme 2 [13]

Soit  $x^k$  le  $k^{\text{ième}}$  itéré de l'algorithme, alors :

$$\frac{c^t x^k}{c^t x^0} \leq (\exp [P(x^k) - P(x^0)])^{1/n} . \square$$

Donc, si la suite  $(P(x^k))$  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  $0 < \alpha \leq \frac{1}{4}$ .

## Théorème 3 [13]

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

## 3.2 Généralisation de l'algorithme de base

Soit le programme linéaire sous forme standard :

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

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

On suppose que :

(H1) La matrice  $A$  est de plein rang ( $rg A = m < n$ ).

(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.

On définit la transformation projective  $T_k$  par :

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

avec :

$$\left\{ \begin{array}{l} y_i = \frac{x_i/x_i^k}{1 + \sum_{i=1}^n x_i/x_i^k}, i = 1 : n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{array} \right.$$

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  $(PL)$ , on obtient :

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

ou encore :

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

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

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

### 3.3 Extension de la méthode de Karmarkar

Considérons le problème d'optimisation non linéaire suivant :

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

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 ( $rgA = m < n$ ).

(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.

#### Formulation du problème

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

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

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 :  $a_i = \frac{1}{n+1}, \forall i \in \{1, \dots, n+1\}$ .  $e_{n+1} \in \mathbb{R}^{n+1}$  où  $e_i = 1, \forall i \in \{1, \dots, n+1\}$ .

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

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

avec :

$$\begin{cases} y_i = \frac{x_i/x_i^k}{1 + \sum_{i=1}^n x_i/x_i^k}, i = 1 : n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{cases}$$

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

Le problème :

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

se transforme comme suit :

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

Maintenant, on considère le problème non linéaire suivant :

$$(2) \begin{cases} \min g(y) = y_{n+1} [f(T_k^{-1}(y)) - z^*] \\ A_k y = 0 \\ y \in S_{n+1} \end{cases}$$

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

Notons que la valeur optimale de  $g$  est 0 et que le centre du simplexe est réalisable pour (2).

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 :

### Lemme 3

*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  $Y = \{y \in \mathbb{R}^{n+1} : A_k y = 0, y \in S_{n+1}\}$ .*

## Preuve

Puisque  $g$  est une fonction continue sur  $Y$ , il suffit de montrer que :

$$g\left(\frac{1}{2}y + \frac{1}{2}\hat{y}\right) \leq \frac{1}{2}[g(y) + g(\hat{y})], \forall y, \hat{y} \in Y.$$

Nous avons :  $\forall y, \hat{y} \in Y, \exists x, \hat{x} \in \mathbb{R}^n$  tel que :  $x = T_k^{-1}(y)$  et  $\hat{x} = T_k^{-1}(\hat{y})$ , et alors :

$$\begin{aligned} g\left(\frac{1}{2}y + \frac{1}{2}\hat{y}\right) &= \frac{1}{2}(y_{n+1} + \hat{y}_{n+1}) \left[ f\left(\frac{D_k\left(\frac{1}{2}y[n] + \frac{1}{2}\hat{y}[n]\right)}{\frac{1}{2}(y_{n+1} + \hat{y}_{n+1})}\right) - z^* \right] \\ &= \frac{1}{2}(y_{n+1} + \hat{y}_{n+1}) \left[ f\left(\frac{y_{n+1}}{y_{n+1} + \hat{y}_{n+1}}x + \frac{\hat{y}_{n+1}}{y_{n+1} + \hat{y}_{n+1}}\hat{x}\right) - z^* \right] \\ g\left(\frac{1}{2}y + \frac{1}{2}\hat{y}\right) &\leq \frac{1}{2}(y_{n+1} + \hat{y}_{n+1}) \left[ \frac{y_{n+1}}{y_{n+1} + \hat{y}_{n+1}}f(x) + \frac{\hat{y}_{n+1}}{y_{n+1} + \hat{y}_{n+1}}f(\hat{x}) - z^* \right] \\ &\leq \frac{1}{2}y_{n+1}f(x) + \frac{1}{2}\hat{y}_{n+1}f(\hat{x}) - \frac{1}{2}(y_{n+1} + \hat{y}_{n+1})z^* \\ &\leq \frac{1}{2}y_{n+1}[f(x) - z^*] + \frac{1}{2}\hat{y}_{n+1}[f(\hat{x}) - z^*] \end{aligned}$$

Donc :  $g\left(\frac{1}{2}y + \frac{1}{2}\hat{y}\right) \leq \frac{1}{2}g(y) + \frac{1}{2}g(\hat{y})$ .  $\square$

## 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 :

$$(3) \left\{ \begin{array}{l} \min \nabla g(a)^t y \\ A_k y = 0 \\ e_{n+1}^t y = 1, y \geq 0 \\ \|y - a\|^2 \leq \alpha^2 \end{array} \right.$$

On sait que pour  $\alpha < 1$  la contrainte  $y \geq 0$  est redondante et donc (3) s'écrit sous la forme :

$$(4) \left\{ \begin{array}{l} \min \nabla g(a)^t y \\ A_k y = 0 \\ e_{n+1}^t y = 1 \\ \|y - a\|^2 \leq \alpha^2 \end{array} \right.$$

**Lemme 4**

La solution optimale du problème (4) est donnée explicitement par :  $y^k = a - \alpha d^k$

où  $d^k = \frac{p^k}{\|p^k\|}$  et  $p^k = p_{B_k}(\nabla g(a))$ ,  $B_k = \begin{bmatrix} A_k \\ e_{n+1}^t \end{bmatrix}$ .

**Preuve**

On pose  $x = y - a$ , alors on a :  $B_k x = \begin{bmatrix} A_k \\ e_{n+1}^t \end{bmatrix} (y - a) = 0$ , et le problème (4) est équivalent à :

$$(5) \left\{ \begin{array}{l} \min \nabla g(a)^t x \\ B_k x = 0 \\ \|x\|^2 \leq \alpha^2 \end{array} \right.$$

$x^*$  est une solution de (5) si et seulement si  $\exists \lambda \in \mathbb{R}^{m+1}$ ,  $\exists \mu \geq 0$  tels que :

$$\nabla g(a) + B_k^t \lambda + \mu x^* = 0 \dots \dots \dots (*)$$

En prèmultipliant les deux membres de(\*) par  $B_k$  et puisque  $B_k x^* = 0$ , on trouve :

$$B_k \nabla g(a) + B_k B_k^t \lambda + \mu B_k x^* = B_k \nabla g(a) + B_k B_k^t \lambda = 0$$

alors  $\lambda = -(B_k B_k^t)^{-1} (B_k \nabla g(a))$ , en substituant dans (\*) :

$$x^* = -\frac{1}{\mu} [I - B_k^t (B_k B_k^t)^{-1} B_k] \nabla g(a) = -\frac{1}{\mu} p_k$$

$$\|x^*\| = \frac{1}{\mu} \|p_k\| = \alpha \Rightarrow x^* = -\alpha \frac{p^k}{\|p^k\|} = -\alpha d^k$$

et on a :  $y^k = y^* = a + x^* = a - \alpha d^k$

Ce qui termine la démonstration.  $\square$

## 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 envoyer 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  $D_k = \text{diag}(x^k)$ ,  $A_k = [AD_k \quad -b]$ ,  $B_k = \begin{bmatrix} A_k \\ e_{n+1}^t \end{bmatrix}$
- 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.**

## Etude de la convergence

Pour établir la convergence de notre algorithme, on introduit la fonction potentiel associée au problème ( $PNL$ ) 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  $(f(x) - z^*)$ .

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

$$\frac{P(x^k) - P(x^0)}{n + 1} = \ln \left[ \frac{f(x^k) - z^*}{f(x^0) - z^*} \right] - \left[ \frac{\sum_{i=1}^n \ln(x_i^k) - \sum_{i=1}^n \ln(x_i^0)}{n + 1} \right]$$

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

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

On a le lemme suivant :

### Lemme 5

$$\text{On a : } g(y^k) < g(a)$$

### Preuve

$$g(y^k) = g(a) + \langle \nabla g(a), y^k - a \rangle \text{ et } y^k = a - \alpha d^k, \text{ donc :}$$

$$\begin{aligned} g(y^k) - g(a) &= \langle \nabla g(a), -\alpha d^k \rangle \\ &= \left\langle \nabla g(a), -\alpha \frac{p^k}{\|p^k\|} \right\rangle = -\frac{\alpha}{\|p^k\|} \langle \nabla g(a), p^k \rangle \end{aligned}$$

$$= -\frac{\alpha}{\|p^k\|} \|p^k\|^2 = -\alpha \|p^k\| < 0$$

Donc  $g(y^k) < g(a)$  .□

La solution  $y^k$  vérifie la propriété suivante :

**Lemme 6**

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

où  $y^k$  est la solution optimale du problème (4).

**Preuve**

Soit  $x^*$  la solution optimale du problème (PNL), et  $y^*$  tel que  $x^* = T_k^{-1}(y^*)$ , on distingue deux cas :

**Cas 1)**  $y^* \in Y \cap B(a, \alpha)$

$$0 \leq g(y^k) \leq g(y^*) = 0 < \left(1 - \frac{\alpha}{n+1}\right) g(a)$$

d'où le résultat.

**Cas 2)**  $y^* \notin Y \cap B(a, \alpha)$

$y^* \notin Fr(B(a, \alpha)) \cap [a, y^*]$ , on a :

$$Fr(B(a, \alpha)) \cap [a, y^*] = \begin{cases} \{y'^k\} & (\text{un point réalisable quelconque}) \\ \{y^k\} & (\text{une solution optimale}) \end{cases}$$

**a)**  $Fr(B(a, \alpha)) \cap [a, y^*] = \{y'^k\}$

$$\exists \theta \in ]0, 1[ : \begin{cases} y'^k = \theta y^* + (1 - \theta)a \\ \|y'^k - a\| = \alpha \end{cases}$$

$$\|y'^k - a\| = \alpha \Leftrightarrow \theta \|y^* - a\| = \alpha$$

$$\|y^* - a\| = \frac{\alpha}{\theta}$$

Et on a :  $\|y^* - a\|^2 = \|y^*\|^2 + \|a\|^2 - 2a^t y^*$

$$\|y^*\|^2 = \sum_{i=1}^{n+1} (y_i^*)^2 \leq \left(\sum_{i=1}^{n+1} y_i^*\right)^2 = (e_{n+1}^t y^*)^2 = 1$$

$$\|a\|^2 = \left\| \frac{1}{n+1} e_{n+1} \right\|^2 = \frac{1}{n+1}$$

$$a^t y^* = \frac{1}{n+1} e^t y^* = \frac{1}{n+1}$$

$$\|y^* - a\|^2 \leq \|y^*\|^2 + \|a\|^2$$

$$\text{Alors : } \|y^* - a\|^2 \leq 1 + \frac{1}{n+1} \leq 1 + (n+1) \leq (n+1)^2$$

$$\text{On a aussi : } \|y^* - a\|^2 = \left(\frac{\alpha}{\theta}\right)^2$$

$$\left(\frac{\alpha}{\theta}\right)^2 \leq (n+1)^2 \Rightarrow \frac{\alpha}{\theta} \leq n+1 \Rightarrow \theta \geq \frac{\alpha}{n+1}$$

Puisque  $g$  est convexe et  $g(y^*) = 0$ , alors :

$$g(y^k) \leq \theta g(y^*) + (1-\theta)g(a) \leq \left(1 - \frac{\alpha}{n+1}\right) g(a)$$

$$\text{De plus, } g(y^k) \leq g(y^k), \text{ donc : } g(y^k) \leq \left(1 - \frac{\alpha}{n+1}\right) g(a)$$

$$\text{b) } Fr(B(a, \alpha)) \cap [a, y^*] = \{y^k\}$$

On trouve le même résultat par une démonstration analogue.  $\square$

#### **Théorème 4**

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

#### **Preuve**

$$\begin{aligned} P(x^{k+1}) - P(x^k) &= (n+1) \ln \left( \frac{f(x^{k+1}) - z^*}{f(x^k) - z^*} \right) - \sum_{i=1}^n \ln \left( \frac{x_i^{k+1}}{x_i^k} \right) \\ &= (n+1) \ln \left( \frac{g(y^k)}{g(a)} \right) - \sum_{i=1}^{n+1} \ln(y_i^k) \\ &\leq (n+1) \ln \left( 1 - \frac{\alpha}{n+1} \right) - \sum_{i=1}^{n+1} \ln(y_i^k) \\ &\leq -\alpha + \frac{\alpha^2}{2(1-\alpha)^2} \end{aligned}$$

On a utilisé le résultat démontré par Karmarkar [9,11] :  $-\sum_{i=1}^{n+1} \ln(y_i^k) \leq \frac{\alpha^2}{2(1-\alpha)^2}$

Donc,  $P(x^{k+1}) \leq P(x^k) - \delta$  où  $\delta = \alpha - \frac{\alpha^2}{2(1-\alpha)^2}$  .  $\square$

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

## *CONCLUSION*

Les méthodes de points intérieurs sont connues par leur efficacité, rapidité de convergence, simplicité algorithmique et capacité de résoudre des problèmes de grandes tailles. L'inconvénient principal dans ce type de méthodes est l'initialisation, c'est-à-dire la détermination d'un point initial qui se trouve à l'intérieur du domaine. La plupart des résultats théoriques supposent que ce point initial est connu, mais numériquement l'obtention de ce point initial prend beaucoup de temps et représente un handicap majeur pour ces méthodes originales et importantes.

Dans cette thèse, nous avons apporté des contributions d'ordre algorithmique, théorique et numérique. En effet, l'introduction du poids et également l'approche non réalisable constituent un remède appréciable pour le problème d'initialisation au niveau des méthodes de trajectoire centrale.

De même, nous avons pu développer deux variantes de méthodes de points intérieurs pour la minimisation d'une fonction convexe sous contraintes linéaires avec établissement de tout l'aspect théorique. Une étude numérique approfondie des algorithmes proposés ainsi que leur généralisation à des programmes non linéaires fera sans doute une étude de recherche intéressante.

# *BIBLIOGRAPHIE*

- [1] M. Achache, " A weighted path-following method for the linear complementarity problem ", *Studia Universitatis Babes-Bolyai. Series Informatica*, 48 (1) 61- 73 (2004).
- [2] M. Achache, " Multidimensional primal-dual path-following interior point methods for linear programming and linear complementarity problems", *Thèse d'état, Université Ferhat Abbas de Sétif, Algérie*, (2005).
- [3] M. Achache, H. Roumili & A. Keraghel, " A numerical study of an infeasible primal-dual path-following algorithm for linear programming ", *Applied Mathematics and Computation*, Volume 186, 2, 1472-1479 (2007).
- [4] S. Bazarra, H.D. Sherali and C.M. Shetty, " Nonlinear programming, theory and algorithms ", Second edition (1993).
- [5] Zs.Darvay, " A weighted-path-following method for linear optimization ", *Studia Universitatis Babes-Bolyai. Series Informatica*, 47 (1) 3-12, (2002).
- [6] Z. Darvay, " New interior point algorithms in linear programming ", *AMO, Advanced Modeling and Optimization*, Volume 5, Number 1 (2003).
- [7] J. Ding and T.Y. Li, " An algorithm based on weighted logarithmic barrier function for linear complementarity problem ", *The Arabian journal for Science and Engineering* V15, N 4B (1990).
- [8] J. Dominguez & M. Gonzalez-Lima, " A primal-dual interior point algorithm for quadratic programming ", *Technical report 01, Venezuela* (2005).
- [9] A. Hanachi, " Etude théorique et numérique d'une méthode de linéarisation de type Karmarkar pour la programmation quadratique convexe ", *Thèse de Magister, Institut de Mathématiques, Université Ferhat Abbas, Sétif* (1997).
- [10] N. Karmarkar, " A new polynomial-time algorithm for linear programming ". *Combinatorica* 4, 373-395 (2005).

- [11] Z. Kebbiche, A. Keraghel & A. Yassine, " Extension of a projective interior point method for linearly constrained convex programming", *Applied Mathematics and Computation* Vol. 193, Issue 2, 553-559, 1 November (2007).
- [12] Z. Kebbiche, A. Keraghel & A. Yassine, " An infeasible interior point method for the monotone linear complementarity problem", *International Journal of Mathematical Analysis*, Vol. 1, No 17, 841-849, 2007.
- [13] A. Keraghel, " Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar ", Thèse de Doctorat de Mathématiques appliquées, Université Joseph Fourier, Grenoble, France (1989).
- [14] A. Keraghel, Z. Kebbiche & M. Achache, " An implementing weighted path-following algorithm for linear complementarity problems", *Analete Universitatii Oradea, Fasc. Matematica*, Tom XIV, 53-64, (2007).
- [15] M. Kojima, S. Mizuno & A. Yoshise, "A polynomial time algorithm for class of linear complementarity problems ", *Mathematical programming* 44, 1-26 (1989).
- [16] B. Merikhi, " Etude comparative de l'extension de l'algorithme de Karmarkar et des méthodes simpliciales pour la programmation quadratique convexe ", Thèse de Magister, Institut de Mathématiques, Université Ferhat Abbas, Sétif (1994).
- [17] B. Merikhi, " Extension de quelques méthodes de points intérieurs pour la programmation semi-définie ", Thèse de Doctorat, Institut de Mathématiques, Université Ferhat Abbas, Sétif (2006).
- [18] A. Nemirovskii, K. Scheinberg, " Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems ", *Mathematical Programming: Series A* 72(3): 273-289 (1996).
- [19] J. Nocedal & S.J. Wright, " Numerical optimization ", Springer series in operations research.

- [20] H. Roumili, " Etude qualitative des méthodes de points intérieurs non réalisables pour la (PL) et la (PQC) ", Thèse de Magister, Institut de Mathématiques, Université Ferhat Abbas, Sétif (1998).
- [21] E.M. Simantiraki & D.F. Shanno, " An infeasible-interior point method for linear complementarity problems ", SIAM J. Optim. Vol. 7, No.3, 620-640 (1997).
- [22] M. J. Todd, B. P. Burell, " An extension of Karmarkar's algorithm for linear programming using dual variables ", Algorithmica 1: 409-424 (1986).
- [23] Y. Wang & P.Fei, " A primal -infeasible interior point algorithm for linearly constrained convex programming ", Optimization-online.org.
- [24] S. J. Wright, " Primal-dual interior point methods", SIAM. Philadelphia, (1997).
- [25] S. J. Wright & Y. Zhang, " A superquadratic-interior-point method for linear complementarity problems ", Mathematical Programming 73, 269-289 (1996).
- [26] Y. Ye & E. Tse, " An extension of Karmarkar's projective algorithm for convex quadratic programming ", Mathematical Programming 44 157-179 (1989).

# *CHAPITRE I*

## *MÉTHODES DE TRAJECTOIRE CENTRALE POUR LE PROBLÈME DE COMPLÉMENTARITÉ LINÉAIRE*

## *CHAPITRE II*

*MINIMISATION D'UNE FONCTION CONVEXE SOUS  
CONTRAINTES LINÉAIRES PAR UNE MÉTHODE DE  
TRAJECTOIRE CENTRALE AVEC POIDS*

## *CHAPITRE III*

*UNE VARIANTE DE TYPE PROJECTIF POUR LA  
MINIMISATION D'UNE FONCTION CONVEXE SOUS  
CONTRAINTES LINÉAIRES*