République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna 2 – Mostefa Ben Boulaid -

Faculté/institut de : Mathématiques et Informatique
Département de : Mathématiques

THESE

Présentée pour l'obtention du titre de :

Docteur en Mathématiques
Option : Mathématiques Appliquées

Sous le thème :

Etude théorique et numérique de quelques méthodes de points intérieurs pour l'optimisation semi-définie.

Présentée par :
LAOUAR Mounia

Devant le jury composé de :

• Dr. Safa Menkad	M.C.A.	Université de Batna 2	Présidente
• Dr. El Amir Djeffal	M.C.A.	Université De Batna 2	Rapporteur
• Dr. Zouhir Mokhtari	Pr.	Université Mohamed Kheidar. Biskra	Examinateur
• Dr. Yahia Djabrane	M.C.A.	UniversitÈ Mohamed Kheidar. Biskra	Examinateur

2021/2022

I dedicate this thesis to :

my late father Abd Elkrim

and

my late mother Zoubida Achouri

who were the first to encourage me to go so far in my studies.

They instilled in me a taste for hard work, rigour and ambition.

Because they always supported me, even in their last days.

Thank you Mum and Dad, thank you for everything,

my beloved son Mohamed,

and

my brothers: Salem, Malik and Redha.

Acknowledgements

No work would be completed without sacrifice, motivation and encouragement.

I owe my deepest gratitude to Almighty Allah Who Gave me such strength and enabled me to accomplish this work successfully.

To complete this work I am greatly thankful from core of my heart to my supervisor Dr. *El Amir DJEFFAL* for his patience, precious guidance and comprehension during the whole work,

I wish to extend my warmest thank to the board of examiners,

- *Safa Menkad*, Dr at Batna2 University,

- *Zouhir Mokhtari* Pr at Biskra University,

- *Yahia Djebrane* Pr at Biskra university,

I would like also to thank all the teachers and the colleagues involved in this study for their contribution,

I owe my loving thanks to my brothers, my late parents and my dear son for their love and support throughout my life,

At Last I offer my regards and blessings to all of those who supported me in any respect during the completion of this work.

Contents

Chapter 1

Introduction

Optimization in science, engineering, economics, and industry, it is essential for students and practitioners alike to develop an understanding of optimization algorithms.Knowledge of the capabilities and limitations of these algorithms leads to a better understanding of their impact on various applications, and points the way to future research on improving and extending optimization algorithms and software.

Dantzig's [7] development of the simplex method in the late 1940s marks the start of the modern era in optimization. This method made it possible for economists to formulate large models and analyze them in a systematic and efficient way. Dantzig's discovery coincided with the evelopment of the first digital computers, and the simplex method became one of the earliest important applications of this new and revolutionary technology. From those days to the present, computer implementations of the simplex method have been continually improved and refined. They have benefited particularly from interactions with numerical analysis, a branch of mathematics that also came into

its own with the appearance of digital computers, and have now reached a high level of sophistication.

Today, linear programming and the simplex method continue to hold sway as the most widely used of all optimization tools. Since 1950, generations of workers in management, economics, finance, and engineering have been trained in the business of formulating linear models and solving them with simplex-based software. Often, the situations they model are actually nonlinear, but linear programming is appealing because of the advanced state of the software, guaranteed convergence to a global minimum, and the fact that uncertainty in the data can make complicated nonlinear models look like overkill. Nonlinear programming may make inroads as software develops, and a new class of methods known as interior-point methods has proved to be faster for some linear programming problems.

In the 1980s it was discovered that many large linear programs could be solved efficiently by formulating them as nonlinear problems and solving them with various modifications of nonlinear algorithms such asNewton's method. One characteristic of these methods was that they required all iterates to satisfy the inequality constraints in the problem strictly, so they soon became known as interior-point methods. By the early 1990s, one class primal–dual methods had distinguished itself as the most efficient practical approach and proved to be a strong competitor to the simplex method on large problems.

The motivation for interior-point methods ($\mathcal{IPMs}$) arose from the desire to find algorithms with better theoretical properties than the simplex method. The simplex method [7] can be quite inefficient on certain problems. Roughly speaking, the time required to solve a linear programmay be exponential in the size $O(2^n)$ of the problem, as measured

by the number of unknowns and the amount of storage needed for the problem data. In practice, the simplex method is much more efficient than this bound would suggest, but its poor worst-case complexity motivated the development of new algorithms with better guaranteed performance. Among them is the ellipsoid method proposed by Khachiyan [19] , which finds a solution in time that is at worst polynomial in the problem size. Unfortunately, this method approaches its worst-case bound on all problems and is not competitive with the simplex method.

Karmarkar's projective algorithm [17], announced in 1984, also has the polynomial complexity property, Padberg [32] proposed a new potential function for the convergence., but it came with the added inducement of good practical behavior. The initial claims of excellent performance on large linear programs were never fully borne out, but the announcement prompted a great deal of research activity and a wide array of methods described by such labels as "affine-scaling," "logarithmic barrier," "potential reduction," "path-following," "primal–dual," and "infeasible interior-point." All are related to Karmarkar's original algorithm, and to the log-barrier approach, but many of the approaches can be motivated and analyzed independently of the earlier methods.

Interior-point methods share common features that distinguish them fromthe simplex method. Each interior-point iteration is expensive to compute and can make significant progress towards the solution and their polynomial complexity appeared in the mid-fifties, while the simplex method usually requires a larger number of inexpensive iterations.The simplex method works its way around the boundaryof the feasible polytope, testing a sequence of vertices in turn until it finds the optimal one. Interior-point methods approach

the boundary of the feasible set only in the limit. They may approach the solution either from the interior or the exterior of the feasible region, but they never actually lie on the boundary of this region.

However, it should be pointed out that (technically speaking), these methods have theoretical and numerical disadvantages, among others: the problem of initialization and the excessive cost of iteration related to management choices and of the step of displacement.

We have made a theoretical, algorithmic and numerical contribution to the solution of problems in particular linear complementarity monotone problems . Our objectif in this thesis is the study of the complexity analysis and the numerical implementation of an interior-point algorithm based on a new kernel function. The algorithm based on the strategy of the central path and on a method for finding a new search directions for the monotone linear complementarity problem, where we show that the short-step algorithm deserves the best current polynomial complexity.

This thesis consists of four chapters and is organised as follows:

In chapter1, to introduce some notions of a convixity and a mathematical programming and we will make a synthesis on convex problems and their solution methods.

The chapter2 we will introduce two families of interior points methods. The primal-dual trajectory methods were introduced in the late 1980s as a variant of the Karmarkar approach and were fully developed in the early Karmarkar approach, and were fully developed in the early 1990s.We use the linear complementarity methods (LCP) for solving the (QP). We will discuss the study of the families of interior-point programming, the most well known in optimization, are the following: The reduction of the projective potential

(Karmarkar's algorithm) and the central path method ($\mathcal{CPM}$).

The third chapter we present an interior-point algorithm for solving an optimization problem using the central path method. By an equivalent reformulation of the central path, we obtain a new search direction which targets at a small neighborhood of the central path. For a full-Newton step interior-point algorithm based on this search direction, the complexity bound of the algorithm is the best known for linear complementarity problem.

The last chapter we present two primal-dual small-step and large-step central path algorithms for solving a monoton linear complementarity program based on kernel function . Without forgetting of course, to conclude our thesis with by a conclusion and prespectives.

Chapter 2

Preliminaries and fondamental concepts

In this chapter we will introduce some notions of a convixity and a mathematical programming and we will make a synthesis on convex problems and their solution methods.

2.1 Convexity concepts

In this section, we present a reminder of the fundamental notions of the convexity properties which will serve as a basis for the following.

Definition 1 *D is a convex set if only if D contains line segment between any two points in the D, such as:*

$$\forall x_1,\ x_2 \in D,\ 0 \leq \theta \leq 1 \Longrightarrow \theta x_1 + (1-\theta)x_2 \in D.$$

Definition 2 *D is a convex polyhedron if it is of the following form:*

$$D = \left\{x \in \mathbb{R}^n : a_i^t x \leq b_i,\ i = \overline{1,m}\right\} ..$$

where a_i are a non-zero vector of $\mathbb{R}^n$ and b_i are a scalar for $i = \overline{1,m}$.

- We can write D under a matix form, as follow:

$$D = \{x \in \mathbb{R}^n : Ax \leq b\}, \text{ where: } A = (a_i^t)_{i=1}^m \in \mathbb{R}^{m\times n} \text{ and } b = (b_i)_{i=1}^m \in \mathbb{R}^m.$$

Definition 3 *S_n is an $n\_simplex$ if it is of the following form:*

$$S_n = \left\{x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\right\}.$$

Definition 4 *A point $x \in S_n$ is considred to be extremal (or vertex of S_n) if we have:*

$$\forall t \in [0,1],\ \forall (y,z) \in S_n^2 :\ x = (1-t)y + tz \Longrightarrow x = y = z.$$

Definition 5 *f is called convex function into the convex subset D if only if the condition hold:*

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

Definition 6 *f is called strictly convex function into the convex subset D if only if the following condition:*

$$\textit{Forall } 0 < \lambda < 1 \textit{ and all } x_1,\ x_2 \in D, x \neq y : f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2).$$

Definition 7 *f is called (strictly) concave over a convex set D if $-f$ is (strictly) convex over D.*

2.1.1 First and second order characterizations of convex functions

Theorem 8 *Suppose $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ is twice differentiable over an open domain. Then, the following are equivalent:*

1. f is convex.
2. $f(y) \geq f(x) + \nabla f(x)^T (y - x)$, forall $x; y \in D$.
3. $\nabla^2 f(x) \geq 0$, forall $x \in D$.

Corollary 9 *Consider an unconstrained optimization problem*

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

where f is convex and differentiable. Then, any point x that satisfies $\nabla f(\bar{x}) = 0$ is a global minimum.

1. $\nabla f(x) = 0$ is always a necessary condition for local optimality in an unconstrained problem. The theorem states that for convex problems, $\nabla f(x) = 0$ is not only necessary, but also sufficient for local and global optimality.
2. In absence of convexity, $\nabla f(x) = 0$ is not sufficient even for local optimality (e.g., think of $f(x) = x^3$ and $x = 0$).
3. Another necessary condition for (unconstrained) local optimality of a point x was $\nabla^2 f(x) \geq 0$. Note that a convex function automatically passes this test.

2.1.2 Strict convexity and uniqueness of optimal solutions

Characterization of Strict Convexity

Recall that a function

$f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is strictly convex if $\forall x; y; x \neq y; \forall \lambda \in (0, 1)$,

$$f(x + (1 - \lambda)y) < f(x) + (1 - \lambda)f(y).$$

Like we mentioned before, if f is strictly convex, then f is convex (this is obvious from the definition) but the converse is not true (e.g., $f(x) = x$; $x \in \mathbb{R}$).

Second order sufficient condition $\nabla^2 f(x) > 0; \forall x \in D \Longrightarrow f$ strictly convex on D.

First order characterization A function f is strictly convex on $\mathbb{R}^n$ if and only if $f(y) > f(x) + \nabla^T f(x)(y - x); \forall x, y \in D, x \neq y$.

There are similar characterizations for strongly convex functions. For example, f is strongly convex if and only if there exists $m > 0$ such that

$$f(y) \geq f(x) + \nabla^T f(x)(y - x) + m \left\| y - x \right\|^2; \forall x; y \in D,$$

or if and only if there exists $m > 0$ such that

$$\nabla^2 f(x) \geq mI, \forall x \in D.$$

Example 10 *The function* $(x, y) \longmapsto f(x, y) = x^2 + y^2 + xy$ *is convex on* $\mathbb{R}^2$, *where* $\nabla f(x, y) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, *we remak that, the sign of the principal minors of the hessian matrix are:* $2 > 0$, $3 > 0$ *in this case we can conclude that,* f *is a convex function.*

Example 11 *The function* $(x,y) \longmapsto f(x,y) = -y^2 - \exp(x)$ *is convex on* $\mathbb{R}^2$, *where* $\nabla f(x,y) = \begin{pmatrix} -\exp(x) & 0 \\ 0 & -2 \end{pmatrix}$, *we remak that, the sign of the principal minors of the hessian matrix are:* $-\exp(x) < 0$, $2\exp(x) > 0$ *in this case we can't conclude anything about* f. *In this situation we calculat* $\nabla(-f(x,y)) = \begin{pmatrix} \exp(x) & 0 \\ 0 & 2 \end{pmatrix}$, *then, we have* $\exp(x) > 0$ *and* $2\exp(x) > 0$, $(-f)$ *is concave so* f *is a convex function.*

2.2 Mathematical programming

Mathematical programming, and more specifically optimization, is aimed at solving problems in which one seeks to determine which of a large number of candidate solutions gives the best performance. Specifically, one seeks to find a solution that satisfies a set of constraints that minimizes or maximizes a given function. The application of mathematical programming is expanding and is found in several fields.

For the problem of minimizing a function $f : \Omega \longrightarrow \mathbb{R}$ over a subset D of the domain $\Omega \subset \mathbb{R}^n$ of the function, we use the notation

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

Here, *s.t.* is an abbrevation for the phrase *subject to the condition.*

The elements of the set D are called the feasible points or feasible solutions of the optimization problem. The function f is the objective function.

Observe that $+\infty$ as a function value of the objective function in a minimization problem.

The *(optimal) value* $z_{\min}$ of the minimization problem is by definition

$$z_{\min} = \begin{cases} \inf\{f(x)/x \in D\} \; if \; D \neq \emptyset, \\ +\infty \qquad\qquad\qquad if \; D = \emptyset. \end{cases}$$

The optimal value is thus a real number if the objective function is bounded below and not identically equal $+\infty$ on the set D, the value is -∞ if the function is not bounded below on D, and the value is $+\infty$ if the objective function is identically equal to $+\infty$ on D or if $D = \emptyset$. In general, D is written as follows

$$D = \left\{x \in \Omega \subset \mathbb{R}^n, \; g_i(x) \leq 0, \forall i = \overline{1,n} \; , \; h_j(x) = 0, \forall j = \overline{1,m}\right\},$$

where g_i, h_j are a function definit to Ω into $\mathbb{R}$.

Our aim in (MP) is to try to find the solution $x \in D$ with the smallest value of the objective function, where it is a feasible set of (MP).

2.2.1 Classification of optimization problem

There are many different optimization algorithms in different scientific applications. However, many methods are only valid for certain types of problems. Thus, it is important to be familiar with the characteristics of the problem in order to identify an appropriate technics for its resolution.

Optimization problems are classified according to the mathematical characteristics of the objective function f, constraints g_i; h_j and optimization variables. There is a particular class of problems which particularly concerns the field of operational research, where the aim is to find the optimal permutation of optimization variables.

Resolution of a mathematical programming (MP)

Definition 12 *A solution* $x^* \in D$ *is a global solution of* (MP) *if* $f(x^*) \leq f(x)$; $\forall x \in \mathbb{R}^n$ *and* $f(x^*)$ *the optimal value.*

Definition 13 *A solution* $x \in D$ *is a local minimum of* (MP) *if* $f(x^*) \leq f(x)$, $\forall x \in D \cap B_\varepsilon(x)$ *where* $B_\varepsilon(x) = \{x \in \mathbb{R}^n : \|x - x^*\| < \varepsilon\}$ *is an open ball of radius* $\varepsilon > 0$ *centred in* x^*.

Remark 14 *Note that the global minimum is not necessarily unique, but the optimal value is.*

Existence and unicity solution

Theorem 15 ***(Weierstrass [9])*** *Let* $D \subset \mathbb{R}^n$ *be compact,and let* $f : D \longrightarrow \mathbb{R}$ *be a continuous function on* D. *Then* f *attaints a maximum and a minimum on* D, *i.e., there exists points* z_1 *and* z_2 *in* D *such that*

$$f(z_1) \geq f(x) \geq f(z_2),\ x \in D.$$

Theorem 16 *If* D *is closed not empty of* $\mathbb{R}^n$, f *is continuous and coercive on* D *(i.e.* $\lim_{\|x\| \to +\infty} f(x) = +\infty$*) then* (MP) *admits at least one optimal solution.*

Theorem 17 *If* D *is non-empty convex of* $\mathbb{R}^n$, f *is strictly convex on* D *then* (MP) *admits an optimal solution at most.*

2.2.2 Conditions of optimality

Why do we need optimality conditions?

In order to analyse or effectively solve an optimization problem, it is fundamental to have the conditions for optimality at one's disposal. In fact, these are not only used to check the validity of the solutions obtained, but often the study of these conditions leads to the development of the resolution algorithms themselves. The approach considered here for obtaining conditions is based on the notions of descent and admissible direction.

The development of optimal conditions in the presence of constraints is based on the same intuition as in the case without constraints: it is impossible to go down from a minimum.

Theorem 18 ***(Karush-Kuhn-Tucker. linear constraints)*** *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ a function differentiable on D, If x is a local minimum of the problem (MP), then it exits a vector $y \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^n_+$ such that:*

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

If, in addition, f; g_i; h_j are convex, the previous conditions are both necessary and sufficient for x to be an global optimum for (MP).

Lagrangian Duality The Lagrangian associated with this problem is the function $L : D \times \mathbb{R}^n_+ \times \mathbb{R}^m \longrightarrow \mathbb{R}$, defined by :

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) + \sum_{i=1}^{m} y_j h_j(x),$$

we put:

$$\alpha(x) = \max_{\lambda,\mu}\{L(x,\lambda,\mu), x \geq 0\}$$

$$= \begin{cases} f(x) \text{ if: } g_i(x) \leq 0, \forall i = 1, ..., n \text{ and } h_j(x) = 0, \forall j = 1, ..., m \\ \\ +\infty \text{ else} \end{cases}.$$

so:

$$\bar{\alpha} = \min_{x \in D} \alpha(x) = \min\{f(x),\ x \in D\}.$$

The dual problem associated with the primal problem is :

$$\bar{\beta} = \max_{(\lambda,\mu)} \min_{x \in D} \{L(x,\lambda,\mu)\}.$$

Where: $-\infty \leq \bar{\beta} \leq \bar{\alpha}$.

1. We can not use the conditions of $\mathcal{KKT}$, if we don't have the qualification of constraints in x^*.

2. If (MP) is a convex program, for x^* is a global minimum the condition of $\mathcal{KKT}$ is necessary and suffisante.

Constraints qualification

If a set D is a convex polyheder (i.e. all constraints are affine functions), then by definition the constraints are qualified at every feasible point.

If a set D is a convex and $\mathring{D} \neq \emptyset$; then the constraints are qualified everywhere, this is Slater's condition.

A constraint of inequality $g_i(x) \leq 0$ is said to be saturated in $x \in D$ if $g_i(x) = 0$.

Remark 19 *The full resolution of* (MP) *deals in order with the following points:*

1. The existence of an optimal solution.

2. The characterization of the solution (this is the optimality conditions).

3. Elaboration of algorithms to calculat this solution.Optimization algorithm

We will present an algorithm to converge towards an optimal solution of the problem (MP). Most constrained optimization algorithms exploit the optimality conditions to determine local minima. We will give here some definitions.

2.2.3 Description

An algorithm is defined by an application A, of D in D, where D is the set of feasible solutions, allowing the generation of an elements sequence of D by the formula:

$$\begin{cases} x_0 \in D, k = 0 \leftarrow \textit{initialization step} \\ \\ x_{k+1} = A(x_k), k = k + 1 \leftarrow \textit{iterations} \end{cases}.$$

We are talking about the algorithm of the interior-points, if instead of using D we will use its interior with $\mathring{D} \neq \emptyset$.

So to define an algorithm is none other than to build a $(x_k)_{k\in\mathbb{N}}$ sequence of D and carry out a study to show its convergence and what we will explain in what it will follow:

2.2.4 Convergence

The algorithm is converged if the $(x_k)_{k\in\mathbb{N}}$ sequence generated by the algorithm converges to a limit x^*.

Convergence rate

The craterea for measuring the speed (or rate) of convergence which is the evolution of the error committed at each iteration

($e_k = \|x_k - x^*\|$).

Definition 20 *Let two functions be* $f; g : \mathbb{N} \longrightarrow \mathbb{R}^+$*:*

1. We note $f(n) = O(g(n))$ when there are integers c and n_0 such that for every $n \geq n_0$: $f(n) \leq cg(n)$.

2. We note $f(n) = \Omega(g(n))$ when there are integers c and n_0 such that for every $n \geq n_0$: $cg(n) \leq f(n)$.

3. We note $f(n) = \Theta(g(n))$, if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

The classification of the speed of convergence is based on the concepts of function comparison in the neighbourhood of $+\infty$. In fact, if we assume that the error e_k does not zero out, the speed of convergence can be:

Linear convergence

If: $\|e_k\| = \Omega(\|e_{k+1}\|)$ and $\frac{\|e_{k+1}\|}{\|e_k\|} \leq 1$, for k sufficiently big. It is also known as the error e_k linearly decreases, i.e. : $\exists c \in]0, 1[, \exists k_0 \in \mathbb{N}, \forall k \geq k_0 : e_{k+1} \leq ce_k$.

2.2.5 Newton's method for solving a non-linear system

Solving (MP) is equivalent to solving the following system of non-linear equations:

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

Let's put:

$$\begin{aligned} F(x,\lambda,y) &= \begin{pmatrix} \nabla f(x) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x) + \sum_{i=1}^{m} y_j \nabla h_j(x) \\ \lambda_i g_i(x), \ i = \overline{1,n} \\ h_j(x), \ j = \overline{1,m} \end{pmatrix} \\ F(x,\lambda,y) &= 0, \end{aligned}$$

where:

$$\begin{aligned} F &: \mathbb{R}^{2n+m} \longrightarrow \mathbb{R}^{2n+m} \\ (x,\lambda,y) &\longmapsto \begin{pmatrix} \nabla f(x) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x) + \sum_{i=1}^{m} y_j \nabla h_j(x) & \lambda_i g_i(x) & h_j(x) \end{pmatrix} \end{aligned}$$

The most popular method applied for the resolution of a non-linear system is Newton's method, in the following we describe its principle. Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ a continuous, differentiable function and $J(x)$ be the Jacobian matrix of the function f. Then we consider

the following non-linear system:

$$f(x) = 0$$

Starting from a vector x^0 of $\mathbb{R}^n$ and using the following formula:

$$x^{k+1} = x^k - J(x^k)^{-1} f(x^k).$$

We construct a sequence of points defined by: $x^{k+1} = x^k + d^k$, where d^k is the direction vector, with: $J(x^k)d^k = -f(x^k)$.

2.2.6 Adimissible direction methods

This class of methods solves a non-linear minimization problem by moving from one point of D to another of its lower cost points. They work according to the following principle: given an x^k element of D, a direction d^k is generated such that for a $\alpha^k > 0$ and sufficiently small, the following properties are ensured :

1. $x^k + \alpha^k d^k$ always belongs to D.

2. $f(x^k + \alpha^k d^k)$ is less than $f(x^k)$.

2.2.7 Linear Programming

The problem of maximization or minimization a linear form over a polyhedron which is given in the form of an intersection of closed halvspaces in $\mathbb{R}^n$, is called *linear programming* abbreviated ($\mathcal{LP}$)**.** The fundamental properties of a linear programming problem are

1. a vector of real variables, whose optimal values are found by solving the problem;

2. a linear objective function;

3. linear constraints, both inequalities and equalities.

Mathematical writing: There are three forms for writing a linear program which are

The canonical form:

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

where $c \neq 0$ is a vector in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, A is an $\mathbb{R}^{m \times n}$ matrix and x is the vector of variables in $\mathbb{R}^n$, also called unknowns or parameters.

The standard form:

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

The general form:

$$\begin{cases} \min c^T x \\ Ax \leq b \\ Bx \geq b' \end{cases},$$

where B is an $\mathbb{R}^{p \times n}$ matrix and b' is a vector in $\mathbb{R}^p$.

One particular formulation of the linear programming problem the standard form is frequently used to describe and analyze algorithms. This form is

$$\begin{cases} \min c^T x \\ Ax = b \quad , \\ x \geq 0 \end{cases} \tag{PL}$$

where c and x are vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $\mathbb{R}^{m \times n}$ matrix. If x satisfies the constraints $Ax = b$, $x \geq 0$, we call it a feasible point; the set of all feasible points is the feasible set.

We can convert any linear program into standard form by introducing additional variables called slack variables and artificial variables into its formulation. Associated with any linear program is another linear program called the dual, which consists of the same data objects arranged in a different way. The dual for (PL) is

$$\begin{cases} \max b^T y \\ A^T y + s = c \quad , \\ s \geq 0 \end{cases} \tag{DL}$$

where y is a vector in $\mathbb{R}^m$ and s is a vector in $\mathbb{R}^n$. Such as

$$\mathcal{F}_{(\mathcal{P})} = \left\{ x \in \mathbb{R}^n : Ax = b, x \geq 0 \right\},$$

$$\mathring{\mathcal{F}}_{(\mathcal{P})} = \left\{ x \in \mathbb{R}^n : Ax = b, x > 0 \right\},$$

et

$$\mathcal{F}_{(\mathcal{D})} = \left\{ (y, z) \in \mathbb{R}^{m+n} : A^T y + s = c, z \geq 0 \right\},$$

$$\mathring{\mathcal{F}}_{(\mathcal{D})} = \left\{(y, z) \in \mathbb{R}^{m+n} : A^T y + s = c, z > 0\right\},$$

the sets of feasible and strictly feasible solutions of the two primal (PL) and dual problems (DL).

We call components of y the dual variables, while s is the vector of a dual slacks. The dual problem could be stated more compactly by eliminating s from (DL) and rewriting the constraints as $A^T y \leq c$. However, it turns out to be expedient for the analysis and implementation of interior-point methods to include s explicitly. The linear problem (PL) often is called the primal, to distinguish it from (DL), and the two problems together are referred to as the primal-dual pair.

Theorem 21 ***(Weak duality)*** *A duality theory that explains the relationship between the two problems (PL) and (DL) has been developed. The feasible set and the solution set for the primal tell us a lot about the dual, and vice versa. For instance, given any feasible vectors x for (PL) and (y, s) for (DL), we have that*

$$b^T y \leq c^T x.$$

Theorem 22 ***(Strong duality)*** *In other words, the dual objective gives a lower bound on the primal objective, and the primal gives an upper bound on the dual. The two objective functions coincide at solutions, so that $b^T y^* = c^T x^*$ whenever x^* solves (PL) and (y^*, s^*) solves (DL).*

We now put a very important theorem that allows us to determine if a pair of vectors, respectively primal and dual feasible, are primal and dual optimal

Theorem 23 ***(Complementary slackness)*** *let x be primal feasible and y be dual feasible. let s is a duality gap vector, where*

$$s = c - A^T y.$$

Then x be primal optimal and (y, s) be dual optimal if and only if $x_i s_i = 0, \forall i = \overline{1, n}$.

2.2.8 Methods of solving a mathematical program

The methods of solving a mathematical program can be classified into three categories:

Gradient methods

Conjugate gradient This method was proposed by Hestenes (1952) to solve a linear system with a positive definite matrix, then generalised by Fletcher and Reeves (1964) to solve a non-linear optimisation problems, it is known for its efficiency in minimising a quadratic function without constraints. In the constrained case, a change of simple variable allows us to return to the case without constraints, in fact: x^0 a point satisfying the constraints ($Ax^0 = 0$) and let $x = x^0 + P_A\ y$ such that$P_A = I - A^T(AA^T)^{-1}A$ is the operator of the projection on the cone of the matrix A. The principle of this method is to progressively construct mutually conjugated directions $d^0, d^1, ..., d^k$ with respect to the Hessian matrix $\nabla^2 f(x)$ of the objective function of the optimization problem. $d^i\ \nabla^2 f(x)\ d^j = 0, \forall i \neq j;\ i,\ j \in \{0, 1, ..., k\}$.

Project gradient (Rosen 1960) The principle of this method is to project at each iteration the gradient on the boundary of the feasible set. It should be noted that this

method is designed for a more general program of the form :

$$\min \{f(x) : Ax = b, x \geq 0\}$$

where f is differentiable not necessarily convex.

Simple method

Among the simplicial methods, the reduced gradient method due to Wolfe. It is a direct extension of the simplex method, applied to quadratic programming. Therefore it has the same drawbacks, i.e. cycling and exponential complexity.

Interior-point algorithm ($\mathcal{IPMS}$)

In conjunction with the methods described above, there are currently interior-point methods available for solving a convex optimization problem. They are extensions of the methods developed for linear programming (affines, projective and central trajectory). Initialization problems, the cost of iteration and the choice of descent direction become more expensive. There are three basic classes of interior-points methods, namely :

Affine Methods This is practically the Karmarkar algorithm without a potential function and no projective transformation, an affine transformation is used and the non-negative constraint is replaced by an ellipsoid containing the new iteration.The algorithm is of a simple structure, unfortunately, it is not easy to demonstrate polynomiality.

Methods of potential reduction The potential function plays a big role in the development of interior-points methods. Karmarkar's algorithm applied to the linear programme in standard form uses a potential function of the form: $(n+1)\log(c^t x -$

$z) - \sum_{i=1}^{n} \log(x_i)$ where z is a lower bound of the value optimal lens. Karmarkar proves the convergence and polynomiality of its algorithm by mounting that this function is reduced at each iteration by at least a constant. Since 1987, researchers have been introducing functions of the potential of the primal-dual type, among which, that of Todd et al [40] defined by : $\phi(x,s) = \rho \log(x^t s) - \sum_{i=1}^{n} \log(x_i s_i)$ pour $\rho > n$, this function played a very important role in the development of potential reduction algorithms after 1988. The algorithms corresponding to these methods have a complexity polynomial, they require $O(\sqrt{n}\,|\log \varepsilon|)$ iterations to reduce the duality step.

Central path methods ($\mathcal{CPM}$) They were introduced at the same time as potential reduction methods and fully developed in the early 1990s. They have good theoretical properties: polynomial complexity and superlinear convergence. The central trajectory algorithms restrict the iterates to a neighbourhood of the central path, the latter is an arc of strictly achievable points. The central trajectory algorithms restrict the iterates to a neighbourhood of the central path, the latter is an arc of strictly achievable points.

Several researchers are trying to generalise the principle of these methods for the non-linear programming. In 1989, Megiddo [27] proposed a primal- dual algorithm with a central path for linear programming with generalization for the linear complementarity problem (LCP). Kojima et al [24] have developed a primal-dual algorithm for linear programming, is an extension for the (LCP) is proposed by the same researchers in 1989 with the complexity $O(\sqrt{n} \log \frac{1}{\varepsilon})$ iterations.

Chapter 3

Some methods of interior-points

In this chapter we will introduce two families of interior points methods. The primal-dual trajectory methods were introduced in the late 1980s as a variant of the Karmarkar approach and were fully developed in the early Karmarkar approach, and were fully developed in the early 1990s.We use the linear complementarity methods (LCP) for solving the (QP). We will discuss the study of the families of interior-point programming, the most well known in optimization, are the following:

1. The reduction of the projective potential (Karmarkar's algorithm).

2. The central path method ($\mathcal{CPM}$).

Therefore, in this chapter our objective is to study them, in order to be able to continue our work.

First of all, we will make a short synthesis on an extension of the Karmarkar's algorithm [17].

3.1 Karmarkar's method

3.1.1 Description of method

In 1984, Karmarkar's proposed a promising algorithm for linear programming, it has attractive theoretical properties and good numerical behaviour.

The simplified Karmarkar program is designed to solve a linear program of the form :

$$\begin{cases} \min c^t x = 0 \\ Ax = 0 \\ x \in S_n = \left\{ x \in \mathbb{R}^n_+ : e_n^t x = 1 \right\} \end{cases} . \qquad (SKP)$$

Where: $x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$ a full row $(rg(A) = m < n)$ and $e_n = (1, 1, ..., 1)^t \in \mathbb{R}^n$.

Suppose that:

1. The optimal value $z^* = 0$ is known .
2. $a = \frac{e_n}{n}$ is a feasible solution.
3. $c^t a \neq 0$.

We have for any general linear program can easily be converted into the following standard form:

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

in the simplified form of Karmarkar (SKP).

Karmarkar [17] uses the following projective transformation:

$$\begin{array}{rcl} \mathcal{T} & : & \mathbb{R}^n_+ \longrightarrow S_{n+1} \\ x & \longmapsto & T(x) = y, \end{array}$$

such that:

$$\mathcal{T}(x) = \begin{cases} (y[n])_i = y_i = \dfrac{\frac{x_i}{a_i}}{1+\sum_{i=1}^{n} \frac{x_i}{a_i}}, \ \forall i = 1, ..., n \\ y_{n+1} = 1 - \sum_{i=1}^{n} y_i \end{cases}.$$

We use the $(\mathcal{T})$ transformation in the problem (PL), afterwards we will have a new linear problem as follows:

$$\begin{cases} \min c'^t y = 0 \\ A'y = 0 \\ y \in S_{n+1} \end{cases}, \qquad (NPL)$$

where: $c' = \begin{bmatrix} Dc \\ -z^* \end{bmatrix} \in \mathbb{R}^{n+1}$, $A' = \begin{bmatrix} AD & -b \end{bmatrix} \in \mathbb{R}^{m\times(n+1)}$ and $D = diag(a)$ where $a = (\frac{1}{n}, ..., \frac{1}{n})^T$.

- a is a feasible strictly solution of the problem (PL).
- $\mathcal{T}$ is an inversible application and:

$$\mathcal{T}^{-1}(y) = x = \frac{Dy[n]}{y_{n+1}}.$$

At each iteration by this transformation is defined:

$$\begin{array}{rcl} \mathcal{T}_k & : & \mathbb{R}^n_+ \longrightarrow S_{n+1} \\ x & \longmapsto & T_k(x) = y, \end{array}$$

where:

$$\mathcal{T}_k(x) = y = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x} \text{ and } \mathcal{T}_k^{-1}(y) = x = \frac{D_k y}{e_n^t D_k y}.$$

with $D_k = diag(x^k)$.

At each iteration, we return to the initial variable x by applying the inverse transformation $\mathcal{T}_k^{-1}$ and so on until the optimality test $c^t x < \varepsilon$ is achieved where the precision given is $\varepsilon > 0$.

The problem transformed from (SKP) by the $\mathcal{T}_k$ transformation is the following linear problem :

$$\begin{cases} \min(D_k c)^t y = 0 \\ AD_k y = 0 \\ y \in S_n = \left\{ y \in \mathbb{R}_+^n : e_n^t y = 1 \right\} \end{cases} .$$

Then:

$$\begin{cases} \min(D_k c)^t y = 0 \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \\ y \in \mathbb{R}_+^{n+1} \end{cases} \qquad (TSKP)$$

In the same way, this transformation permits to place at each iteration, the iteration x^k in the center of the simplex S_n; i.e.; $T_k(x^k) = \frac{e_n}{n}$.

Remark 24 *it is suggested to work to return the general problem* (PL) *to the reduced form* $(TSKP)$ *and then return at each iteration to the initial problem* (PL) *to test the optimality of the iterations.*

Lemma 25 *If we known a feasible solution* y^0 *to linear problem, such that* $\forall i = 1, ..., n+1 : y_i^0 > 0$, *so the ellipsoid:*

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

Proof. We suppose that, $\exists\ j \in \{1, ..., n+1\} :\ y_j^0 \leq 0$ and $\sum_{i=1}^{n+1} (\frac{y_i - y_i^0}{y_i^0})^2 \leq \beta^2$, $0 < \beta < 1$ then: $\sum_{i=1}^{n+1} (\frac{y_i - y_i^0}{y_i^0})^2 = \sum_{j \neq i=1}^{n+1} (\frac{y_i - y_i^0}{y_i^0})^2 + (\frac{y_j - y_j^0}{y_j^0})^2 \geq (\frac{y_j - y_j^0}{y_j^0})^2 \geq 1 > \beta^2$. ■

Before applying the conditions of optimality, Karmarkar [17] relaxes the problem $(TSKP)$ to $(TSKPr)$, so we have to use the last lemma on the the problem $(TSKP)$ and we obatin a new problem $(TSKPr)$:

$$\begin{cases} \min(D_k c)^t y = 0 \\ \\ \begin{bmatrix} AD_k \\ \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} \ . \\ \\ \left\| y - \frac{e_{n+1}}{n+1} \right\| \leq \alpha r \end{cases} \qquad (TSKPr)$$

With the optimality conditions, we can have the analytic form to the optimal solution of the last problem, as follow:

$$y = \frac{e_{n+1}}{n+1} - \alpha r d_k,$$

where:

1. $r = \frac{1}{\sqrt{n(n-1)}}$ and $0 < \alpha < 1$,

2. $B_k = \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix}$,

3. $p_k = p_{B_k}(D_k c)$,

4. $d_k = \frac{p_k}{\|p_k\|}$.

Proof. see [18] ■

The Karmarkar's algorithm of $(\mathcal{PL})$

Begin algorithm
Initialization:
$x^0 = a^0 = \frac{1}{n} e_n$,
$k = 0$
  While $(c^t x^k > \varepsilon)$ do
    Step0
      $D_k = diag(x^k) \in \mathbb{R}^{n \times n}$
      $A_k = AD_k \in \mathbb{R}^{m \times n}$
      $B_k = \begin{pmatrix} A_k \\ ---- \\ e_n^t \end{pmatrix} \in \mathbb{R}^{(m+1) \times n}$
    Step1 $p_k = (I_n - B_k^t (B_k B_k^t)^{-1} B_k) D_k c \in \mathbb{R}^n$
    Step2 $d_k = \frac{p_k}{\|p_k\|}$
    Step3 $y^k = a^0 - \alpha r d_k$, $r = \frac{1}{\sqrt{n(n-1)}}$, $0 < \alpha < 1$
    Step4 $x^{k+1} = \frac{D_k y^k}{e_n^t D_k y^k}$, $k = k + 1$
  End While
End algorithm

Remark 26 *The most expensive operation in the algorithm is to calculate the inverse of* $B_k B_k^t$.

Remark 27 *In general, we find a problem in calculating* $(B_k B_k^t)^{-1}$ *so one of the following methods can be used:*

1. $p_k = (I_n - B_k^t(B_kB_k^t)^{-1}B_k)D_kc = (I_n - A_k^t(A_kA_k^t)^{-1}A_k - \frac{1}{n}e_ne_n^t)D_kc.$

2. Change of variable .-ie-. we put $(B_kB_k^t)^{-1}B_kD_kc = M.$

3.1.2 Convergence study (potential function)

For obtaining the convergence of the algorithm, it has to show that: $\frac{c^tx^{k+1}}{c^tx^k} < q_0$, where $0 < q_0 < 1$ is independent of k. However, it's difficult to find the value of q_0, that is why, Karmarkar to be associated with the objective function c^tx the potential function:

$$f(x) = \sum_{i=1}^{n} \log \frac{c^tx}{x_i} \; defined \; into \; \left\{x \in \mathbb{R}^n : Ax = b, \; e_n^tx = 1 \; and \; x > 0\right\}.$$

The relation between the reduction of the objective function and potential function is:

Lemma 28 *The iteration k in the algorithm, to verify:*

$$\frac{c^tx^k}{c^tx^0} \leq (\exp(f(x^k) - f(x^0)))^{\frac{1}{n}} \; where \; x^0 = a^0 = \frac{e_n}{n}.$$

Proof. see [18, 17] ■

Theorem 29 *If $0 < \alpha \leq \frac{1}{4}$, we begin by $x^0 = \frac{e_n}{n}$ after $O(nq + n\log n)$ iterations to find the feasible point x such that:*

1. $c^tx = 0$

2. $\frac{c^tx}{c^tx^0} \leq 2^{-q}$ where q is a precision set.

On the other hand, Padberg [32] proposed a new potential function:

$$h(x) = \frac{c^tx}{(\prod_{i=1}^{n} x_i)^{\frac{1}{n}}},$$

with the number of iterations is $O(nq)$ for $0 < \alpha < 0.7968...$

3.2 Primal-dual methods

We recall that, the linear programming problem in standard form; that is

$$\begin{cases} \min c^T x \\ Ax = b \quad , \\ x \geq 0 \end{cases} \tag{PL}$$

where c and x are vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $\mathbb{R}^{m\times n}$ matrix. The dual problem for (DL) is

$$\begin{cases} \max b^T y \\ A^T y + s = c \quad , \\ s \geq 0 \end{cases} \tag{DL}$$

where y is a vector in $\mathbb{R}^m$ and s is a vector in $\mathbb{R}^n$.

3.2.1 The optimality conditions ($\mathcal{KKT}$)

The primal–dual solutions of (PL), (DL) are characterized by the Karush–Kuhn–Tucker conditions, which we restate here as follows:

$$\begin{cases} A^T y + s = c \\ Ax = b \\ x_i s_i = 0, i = \overline{1,n} \\ (x,s) \geq 0 \end{cases} . \tag{2.1}$$

Primal–dual methods find solutions (x^*, y^*, s^*) of this system by applying variants of Newton's method to the three equalities in (2.1) and modifying the search directions and step lengths so that the inequalities $(x, s) \geq 0$ are satisfied strictly at every iteration. The

third first equations of the system (2.1) are only mildly nonlinear and so are not difficult to solve by themselves. However, the problem becomes much more difficult when we add the nonnegativity requirement the third equation of (2.1). The nonnegativity condition is the source of all the complications in the design and analysis of interior-point methods. To derive primal–dual interior-point methods, we restate the optimality conditions (2.1) in a slightly different form by means of a mapping F from $\mathbb{R}^{2n+m}$ to $\mathbb{R}^{2n+m}$:

$$F(x,y,s) = \begin{pmatrix} A^T y + s - c \\ Ax - b \\ XSe \end{pmatrix} = 0, \tag{2.2a}$$

where:

$$(x,s) \geq 0, \tag{2.2b}$$

$$X = diag(x_1, x_2, ..., x_n), S = diag(s_1, s_2, ..., s_n), \tag{2.2c}$$

and $e = (1, 1, ..., 1)^T$. Primal–dual methods generate iterates (x^k, y^k, s^k) that satisfy the bounds (2.2b) strictly, that is, $x^k > 0$ and $s^k > 0$. This property is the origin of the term interior-point. By respecting these bounds, the methods avoid spurious solutions, that is, points that satisfy $F(x, y, s) = 0$ but not $(x, s) \geq 0$. Spurious solutions abound, and do not provide useful information about solutions of (PL) or (DL), so it makes sense to exclude them altogether from the region of search.

Many interior-point methods actually require the iterates to be strictly feasible; that is, each (x^k, y^k, s^k) must satisfy the linear equality constraints for the primal and dual problems. If

we define the primal–dual feasible set $\mathcal{F}$ and strictly feasible set $\mathring{\mathcal{F}}$ by

$$\mathcal{F} = \{(x, y, s)/Ax = b, A^T y + s = b, (x, s) \geq 0\}, \tag{2.3a}$$

$$\mathring{\mathcal{F}} = \{(x, y, s)/Ax = b, A^T y + s = b, (x, s) > 0\}, \tag{2.3b}$$

the strict feasibility condition can be written concisely as

$$(x^k, y^k, s^k) \in \mathring{\mathcal{F}}.$$

Like most iterative algorithms in optimization, primal–dual interior-point methods have two basic ingredients: a procedure for determining the step and a measure of the desirability of each point in the search space. As mentioned above, the search direction procedure has its origins in Newton's method for the nonlinear equations (2.2a). Newton's method forms a linear model for F around the current point and obtains the search direction $(\Delta x, \Delta y, \Delta s)$ by solving the following system of linear equations:

$$J(x, y, s) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = -F(x, y, s),$$

where J is the Jacobian of F. If the current point is strictly feasible (that is, $(x, y, s) \in \mathring{\mathcal{F}}$), the Newton step equations become

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix}. \tag{2.4}$$

A full step along this direction usually is not permissible, since it would violate the bound $(x, s) \geq 0$. To avoid this difficulty, we perform a line search along the Newton direction so that the new iterate is

$$(x, y, s) + \alpha(\Delta x, \Delta y, \Delta s),$$

for some line search parameter a $\alpha \in (0, 1]$. Unfortunately, we often can take only a small step along the direction ($\alpha << 1$) before violating the condition $(x, s) > 0$; hence, the pure Newton direction (2.4) often does not allow us to make much progress toward a solution. Primal-dual methods modify the basic Newton procedure in two important ways:

1. They bias the search direction toward the interior of the nonnegative orthant $(x, s) \geq 0$ so that we can move further along the direction before one of the components of (x, s) becomes negative.

2. They keep the components of (x, s) from moving "too close" to the boundary of the nonnegative orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.

3.2.2 The Central Path method ($\mathcal{CP}$)

The central path method ($\mathcal{CP}$) is an arc of strictly feasible points that plays a vital role in the theory of primal-dual algorithms. It is parametrized by a scalar $\tau > 0$, and each

point $(x_\tau, y_\tau, s_\tau) \in \mathcal{T}_C$ solves the following system:

$$\begin{cases} A^T y + s = c \\ Ax = b \\ x_i s_i = \tau, i = \overline{1,n} \\ (x,s) > 0 \end{cases}, \tag{2.5}$$

These conditions differ from the $\mathcal{KKT}$ conditions only in the term τ on the right-hand side of the third equation of the system (2.5). Instead of the complementarity condition the third equation of the system (2.5), we require that the pairwise products $x_i s_i$ have the same value for all indices i. From (2.5), we can define the central path as

$$\mathcal{T}_C = \{(x_\tau, y_\tau, s_\tau)/\tau > 0\}.$$

Another way of defining $\mathcal{T}_C$ is to use the notation introduced in (2.1) and write

$$F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, (x_\tau, s_\tau) > 0. \tag{2.6}$$

We show in next chapter that (x_τ, y_τ, s_τ) is defined uniquely for each $\tau > 0$ if and only if $\mathring{\mathcal{F}}$ is nonempty. Hence, the entire path $\mathcal{T}_C$ is well defined.

The equations (2.5) approximate (2.1) more and more closely as τ goes to zero. If $\mathcal{T}_C$ converges to anything as $\tau \downarrow 0$, it must converge to a primal-dual solution of the linear program. The central path thus guides us to a solution along a route that steers clear of spurious solutions by keeping all the pairwise products $x_i s_i$ strictly positive and decreasing them to zero at the same rate.

Most primal-dual algorithms take Newton steps toward points on $\mathcal{T}_C$ for which $\tau > 0$, rather than pure Newton steps for F. Since these steps arebiased toward the interior of the nonnegative orthant defined by $(x, s) \geq 0$, it usually is possible to take longer steps along them than along the pure Newton steps for F before violating the positivity condition. To describe the biased search direction, we introduce a centering parameter $\sigma \in [0, 1]$ and a duality measure μ, defined by

$$\mu = \frac{\sum_{i=1}^{n} x_i s_i}{n} = \frac{x^T s}{n}. \tag{2.7}$$

which measures the average value of the pairwise products $x_i s_i$. The generic step equations are then

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{bmatrix}. \tag{2.8}$$

The step $(\Delta x, \Delta y, \Delta s)$ is a Newton step toward the point $(x_{\sigma\mu}, y_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{T}_C$, at which the pairwise products $x_i s_i$ are all equal to $\sigma\mu$. In contrast, the step (2.4) aims directly for the point at which the $\mathcal{KKT}$ conditions (2.1) are satisfied. If $\sigma = 1$, the equations (2.8) define a centering direction, a Newton step toward the point $(x_\mu, y_\mu, s_\mu) \in \mathcal{T}_C$, at which all the pairwise products $x_i s_i$ are identical to μ. Centering directions are usually biased strongly toward the interior of the nonnegative orthant and make little, if any, progress in reducing μ. However, by moving closer to $\mathcal{T}_C$, they set the scene for substantial progress on the next iteration. (Since the next iteration starts near $\mathcal{T}_C$, it will be able to take a relatively long step without leaving the nonnegative orthant.) At the other extreme, the value $\sigma = 0$ gives the standard Newton step (2.4), sometimes known as the affine-scaling direction for reasons

to be discussed later. Many algorithms use intermediate values of a from the open interval $(0, 1)$ to trade off between the twin goals of reducing μ and improving centrality.

3.3 Transformation of (CQP) to (LCP)

Since its appearance, complementarity has attracted the interest of several researchers, its importance can be measured by the crucial role it plays in the solution of several problems in different fields: linear programming, convex quadratic program, variational inequalities, mechanics, ...

The main idea of this method is based on replacing a linear complementarity problem by a convex quadratic program. The linear complementarity writes inder the form:

$$\begin{cases} \text{to find } x, y \in \mathbb{R}^n : \\ y = Mx + q \\ x^t y = 0 \\ (x, y) \geq 0 \end{cases}, \qquad (LCP)$$

with: $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$.

We have the (CQP) :

$$\begin{cases} Minimize\ c^t x + \frac{1}{2} x^t Q x \\ Subject\ to\ Ax \leq b, x \geq 0 \end{cases}, \qquad (\text{CQP})$$

where: $Q \in S_+^n$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ $(rgA = m \leq n)$ and $b \in \mathbb{R}^m$.

We have (CQP) is a convex problem with a linear constraints, then the conditions of $\mathcal{KKT}$ are necessary and sufficient so:

$x \in \mathbb{R}^n_+$ is an optimal solution if only if $\exists y \in \mathbb{R}^m_+$, $\lambda \in \mathbb{R}^n_+$ such that:

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

$$\Longleftrightarrow \begin{cases} \lambda = c + Qx + A^T y \\ v = b - Ax \\ y^t v = 0, \lambda^t x = 0 \\ y \geq 0, \lambda \geq 0 \end{cases}$$

$$\Longleftrightarrow \begin{cases} find\ z \in \mathbb{R}^{n+m} : \\ w = Mz + q \\ z^t w = 0 \\ w \geq 0, z \geq 0 \end{cases},$$

where: $w = \begin{bmatrix} \lambda \\ v \end{bmatrix} \in \mathbb{R}^{n+m}$, $M = \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}$,

$z = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^{n+m}$and $q = \begin{pmatrix} c \\ b \end{pmatrix} \in \mathbb{R}^{n+m}$.

In general case, we can't transform a (LCP) to (CQP) exeption if we have $(MLCP)$ .ie. the matrix M is a positive semidefinite. Kojima et al [24] proposed an extension of the center path methods for linear programming.

Theorem 30 *Let us consider the monotone linear complementarity problem:*

$$To\ find\ x \in \mathbb{R}^n,\ y = Mx + q \geq 0\ and\ x^t y = 0, \qquad \text{(MLCP)}$$

- the feasible set is denote of the problem $(MLCP)$:

$$\mathcal{S} = \left\{(x, y) \in \mathbb{R}^{2n} : \ y = Mx + q \geq 0, \ x \geq 0, \ y \geq 0\right\};$$

- the strictly feasible set is denote of the problem $(MLCP)$:

$$\mathring{\mathcal{S}} = \{(x, y) \in S : \ x > 0, \ y > 0\};$$

- the solution set of the problem $(MLCP)$:

$$\Omega = \left\{(x, y) \in S : \ x > 0, \ y > 0 \text{ and } x^t y = 0\right\}.$$

We assume that the following assumptions hold:

1. The strictly feasible $\mathring{\mathcal{S}}$ set is not empty.

2. The matrix M is a positive semidefinite matrix.

These assumptions imply that $\mathring{\mathcal{S}}$ is the relative interior of $\mathcal{S}$ and Ω is nonempty polyhedral convex and bounded set. In addition, $(MLCP)$ is aquivalent to the following convex problem, see, e.g. [44].

$$\min \left\{x^t y : \ y = Mx + q, \ x \geq 0, \ y \geq 0\right\} \qquad (CQP)$$

The notion of central path can be introduced by means of a logarithmic barrier function, it is sufficient to associate to (CQP) the following barrier problem:

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

If $\mu > 0$; the solution of (CQP_μ) converges to the solution of (CQP), so we solve a sequence of (CQP_μ) problems by decreasing the values of μ until we obtain a solution of (CQP).

The general idea of central path methods is to follow a particular path (called the center path or central path), taking the Newtonian direction of travel as the direction of travel. In other words: the algorithm generates a strictly feasible sequence $(x^k; y^k)$ and a sequence (μ^k) which expresses the complementarity condition, is checked when (μ^k) tending to zero.

Proposition 31 *f_μ is a strictly convex function.*

Theorem 32 *M is a positive semidefinite matrix and $\mathring{S} \neq \emptyset$, the problem (CQP_μ) has only one optimal solution forall $\mu > 0$.*

Proof. see [24]. ■

Theorem 33 *Let $\mu > 0$ and M be a positive semidefinite matrix.*

$(x, y) > 0$ is an optimal solution of the problem (CQP_μ) if only if (x, y) satisfy the system:

$$\begin{cases} XYe - \mu e = 0 \\ y = Mx + q \quad , \\ (x, y) > 0 \end{cases} \qquad (S_\mu)$$

then, to solve the problem (CQP_μ) is equivalent to solve the system (S_μ).

Proof. We know that, (CQP_μ) is a convex and differentiable program with affine constraints, so the constraints qualifications, then the $\mathcal{KKT}$ conditions are necessary and

sufficient and write as follow:

$$\begin{cases} Xy - \mu e + XM^T w = 0 \\ Yx - \mu e - Yw = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

$$\Longleftrightarrow \begin{cases} (XM^T + Y)w = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

We have, X, M and Y are positive definite so $XM^T + Y$ is positive definite then $w = 0$, we have the new system:

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

■

Definition 34 *The solution of the problem* (S_μ) *for* $\mu > 0$ *is* $(x_\mu; y_\mu)$. $\mathcal{T}_C = \{(x_\mu; y_\mu), \mu > 0\}$ *is the set of all solutions of the system* (S_μ) *is called the central path.*

The solution of the problem $(\mathbf{S}_\mu)$ Since the first equation of the system $(\mathcal{S}_\mu)$ is non-linear, then we use the Newton method for solving the equation non-linear

$$F_\mu(x, y) = \begin{pmatrix} XYe - \mu e \\ y - Mx - q \end{pmatrix} = 0 \qquad (S'_\mu)$$

we obtain:

$$F_\mu(x,y) + \triangledown F_\mu(x,y).(\Delta x, \Delta y) = 0$$

$$\Longleftrightarrow \begin{pmatrix} XYe - \mu e \\ y - Mx - q \end{pmatrix} + \begin{pmatrix} Y\Delta x + X\Delta y \\ \Delta y - M\Delta x \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Longleftrightarrow \begin{cases} Y\Delta x + X\Delta y = \mu e - XYe \\ \Delta y - M\Delta x = 0 \end{cases},$$

the solution of the equation (S'_μ) is the solution of the linear system:

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

For calculating the new iteration we use the solution of the equation (S'_μ):

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

Centralisation factor

The quality of each solution found is measured by a factor called centrality, is defined by a scalar as follow:

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

If (x, y) is a point in the central path we have: $\delta(x, y, \mu) = 0$.

We would like to control our approximation of the path, This results is the definition of the neighborhood of the center path $T(\theta)$, when we call any point is a neighborhood of the set:

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

The Central path algorithm

Begin algorithm
Initialization:
$\varepsilon > 0$ is a precision's parameter, $0 < \theta \leq 0.1$ is a constant,
$\delta = \frac{\theta}{1-\theta}$
$k = 0$
$(x^0, y^0) \in T(\theta)$
While $((x^k)^t y^k > \varepsilon)$ do
Step1 **calculate the parameters**
$\mu^k = (1 - \frac{\delta}{\sqrt{n}})\frac{(x^k)^t y^k}{n}$
Step2
the newton's direction $(\Delta x^k, \Delta y^k)$ **is a solution of the system** $\begin{cases} Y^k \Delta x^k + X^k \Delta y^k = \mu^k e - X^k Y^k e \\ \Delta y^k - M \Delta x^k = 0 \end{cases}$
Step3
the iteration $(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\Delta x^k, \Delta y^k)$.
$k = k + 1$
End While
End algorithm

3.3.1 Convergence

the new point (x^{k+1}, y^{k+1}) remains close to the central path, more precisely if a reasonable value of the parameter $\mu > 0$.

Theorem 35 *(Z. Kebbiche [21])Let* $0 < \theta \leq 0.1$ *and* $\delta = \frac{\theta}{1-\theta}$, *suppose that:* $(x^k, y^k) \in T(\theta)$ *and*

$\mu^k = (1 - \frac{\delta}{\sqrt{n}})\frac{(x^k)^t y^k}{n}$, *then the point* (x^{k+1}, y^{k+1}) *defined by:*

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

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

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

Chapter 4

A primal-dual interior-point method based on a new kernel function for linear complementarity problem

In this chapter, we present an interior-point algorithm for solving an optimization problem using the central path method. By an equivalent reformulation of the central path, we obtain a new search direction which targets at a small neighborhood of the central path. For a full-Newton step interior-point algorithm based on this search direction, the complexity bound of the algorithm is the best known for linear complementarity problem [10].

4.1 Presentation of the problem

Let us consider the linear complementarity problem (LCP): find vectors x and y in real space $\mathbb{R}^n$ that satisfy the following conditions:

$$x \geq 0,\ y = Mx + q \geq 0 \text{ and } x^t y = 0, \tag{LCP}$$

where q is a given vector in $\mathbb{R}^n$ and M is a given $\mathbb{R}^{n\times n}$ real matrix, where:

- the feasible set is denote of the problem (LCP) :

$$S = \left\{(x,y) \in \mathbb{R}^{2n} :\ y = Mx + q \geq 0,\ x \geq 0,\ y \geq 0\right\};$$

- its strictly feasible set is denote of the problem:

$$S_{str} = \left\{(x,y) \in S :\ x > 0,\ y > 0\right\};$$

- and the solution set of the problem (LCP):

$$\Omega = \left\{(x,y) \in S :\ x > 0,\ y > 0 \text{ and } x^t y = 0\right\}.$$

We assume that the following assumptions hold:

1. The strictly feasible S_{str} set is not empty.

2. The matrix M is a positive semidefinite matrix.

These assumptions imply that S_{str} is the relative interior of S and Ω is nonempty polyhedral convex and bounded set. In addition, (LCP) is aquivalent to the following convex problem, see, e.g. [44].

$$\min\left\{x^t y :\ y = Mx + q,\ x \geq 0,\ y \geq 0\right\} \tag{QP}$$

Hence, finding the solution of (LCP) is equivalent to find the minimizer of (QP) with its objective value is zero. In order to introduce an interior-point method to solve (QP), we associate with it the following barrier minimization problem:

$$\min \left\{ f_\mu(x,y) = x^t y - \mu \sum_{i=1}^{n} \ln(x_i y_i) : \ y = Mx + q, \ x > 0, \ y > 0 \right\}, \tag{3.1}$$

where $\mu > 0$ is a positive real number and it is called the barrier parameter. The problem (3.1) is a convex optimization problem and then its first-order optimality conditions are:

$$\begin{cases} Mx + q = y \\ \\ xy = \mu e, \\ \\ x > 0, \ y > 0 \end{cases} \tag{3.2}$$

If the assumptions (1) and (2) hold then for a fixed $\mu > 0$, the problem (3.1) and the system (3.2) have a unique solution (see [25]) denoted as $(x(\mu), y(\mu))$, with $x(\mu) > 0$ and $y(\mu) > 0$. We call $(x(\mu), y(\mu))$, with $\mu > 0$, the $\mu-$center of (3.1) or (3.2). The set of the $\mu-$centers (with μ running through all positive real numbers) gives a homotopy path, which is called *central path* of (LCP). In the next section, we introduce a method for tracing the central path based a new class of search directions.

4.2 New Class of Search Directions

Now, following [5]. the basic idea behind this approach is to replace the nonlinear equation:

$$\frac{xy}{\mu} = e,$$

in (3.1) by an equivalent equation:

$$\psi(\frac{xy}{\mu}) = \psi(e),$$

where ψ is a real-valued function on $[0, \infty)$ and differentiable on $[0, \infty)$ such that $\Psi(t)$ and $\Psi'(t) > 0$, for all $t > 0$. Then the system (4.2) can be written as the following equivalent form:

$$\begin{cases} Mx + q = y, \\ \\ \psi(\frac{xy}{\mu}) = \psi(e), \\ \\ x > 0, \ y > 0. \end{cases} \tag{3.3}$$

Suppose that we have $(x, y) \in S_{str}$. Applying Newton's method for the system (3.3), we obtain a new class of search directions:

$$\begin{cases} M\Delta x = \triangle y, \\ \\ \frac{y}{\mu}\psi'(\frac{xy}{\mu})\Delta x + \frac{x}{\mu}\psi'(\frac{xy}{\mu})\Delta y = \psi(e) - \psi(\frac{xy}{\mu}). \end{cases} \tag{3.4}$$

Now, the following notations are useful for stadying the complexity of the algorithm. The vectors:

$$v = \sqrt{\frac{xy}{\mu}} \text{ and } d = \sqrt{\frac{xy^{-1}}{\mu}},$$

and observe that these notations lead to

$$\frac{d^{-1}x}{\mu} = dy = v. \tag{3.5}$$

Denote by

$$d_x = \frac{d^{-1}}{\mu}\Delta x,\ d_y = d\Delta y,$$

and hence, we have

$$v(d_x + d_y) = y\Delta x + x\Delta y \tag{3.6}$$

and

$$d_x d_y = \Delta x \Delta y.$$

So using (3.5) and (3.6), the system (3.4) becomes

$$\begin{cases} \overline{M}d_x = d_y, \\ \\ d_x + d_y = Pv, \end{cases}$$

where $\overline{M} = DMD$ with $D = diag(d)$ and

$$P_v = \frac{\psi(e) - \psi(v^2)}{2v\psi'(v^2)} = \frac{-\psi(v^2)}{2v\psi'(v^2)}.$$

The equation $d_x + d_y = Pv$ is called the *scaled centring equation.* It states that the sum of scaled search directions d_x and d_y is equal to $-\nabla\Psi(v)$, the steepest descent direction of $\Psi(v)$.

As in [3], we shall consider the following function:

$$\Psi(t) = (m+1)t^2 - (m+2) + m\frac{1}{t^m},\ \text{for all } t > 0, m \geq 4.$$

Hence, the Newton directions in (3.4) are

$$\begin{cases} M\Delta x = \Delta y, \\ \\ \frac{y}{\mu}\Delta x + \frac{x}{\mu}\Delta y = \frac{-(m+1)v^4+(m+2)v^2-\frac{1}{v^{2m}}}{4(m+1)v^3-2(m+2)v-2m\frac{1}{v^{2m+1}}}. \end{cases} \tag{3.7}$$

4.3 The generic interior-point algorithm for LCP

We can now describe the algorithm in a more formal way. The generic form of the algorithm is shown in the next. The description that closeness of (x, y) to $(x(\mu), y(\mu))$ is measured by the value of $\Psi(t)$, with τ as a threshold value: if $\Psi(t) \leq \tau$, then we start a new outer iteration by performing a $\mu - update$; otherwise we enter an *inner iteration* by computing the search directions at the current iterates with respect to the current value of μ and apply $x^+ = x + \alpha\Delta x$, $y^+ = y + \alpha\Delta y$ to get new iterates. a new kernel function which is defined in the previous section and assume that $\tau \geq 1$. The new interior-point algorithm works as follows. Assume that we are given a strictly feasible point (x, y) which is in a $\tau-$neighborhood of the given $\mu - center$. Then we decrease μ to $\mu^+ = (1 - \theta)\mu$, for some fixed $\theta \in (0, 1)$ and then we solve the Newton system (3.4) to obtain the unique search direction. The positivity condition of a new iterate is ensured with the right choice of the step size α which is defined by some line search rule. This procedure is repeated until we find a new iterate (x^+, y^+) that is a $\tau-$neighborhood and the $\mu^+ - center$. Then μ is again reduced by the factor $1 - \theta$ and we solve the Nweton system targeting at the new $\mu^+ - center$, and so on. this process is repeated until μ is small enough, i.e. $n\mu \leq \varepsilon$. The

parameters parameters τ, θ and the step size α should be chosen in such a way that the algorithm is optimizedin the sense that the number of iterations required by the algorithm is as small as possible. The choice of the so-called barrier update parameter θ plays an important role in both theory and practice of ($\mathcal{IPMs}$). The algorithm for our ($\mathcal{IPMs}$) for the (LCP) is given as follows:

The generic interior-point algorithm for LCP

Begin algorithm
Input:
an accuracy parameter $\varepsilon > 0$, an update parameter θ, $0 < \theta < 1$,
a threshold parameter τ, $0 < \tau < 1$,a strictly feasible point (x^0, y^0)
and $\mu^0 = \frac{(x^0)^t y^0}{n}$ such that $\delta(x^0, y^0, \mu^0) \leq \tau$. begin $x := x^0$, $y := y^0$, $\mu := \mu^0$,
begin
  $\mu = (1 - \theta)\mu$
  While $(\delta(x^0, y^0, \mu^0) > \tau)$ do
    begin
      Solve system (3.4) to obtain $(\Delta x, \Delta y)$,
      Determine a step size α
      $x := x + \Delta x$ $y := y + \Delta y$
  End While
End algorithm

4.4 Kernel functions

To simplify matters we will restrict ourselves in this paper to the case where $\Psi(v)$ is separable with identical coordinate functions. Thus, letting ψ denote the function on the coordinats, we have

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i), \tag{3.8}$$

where $\psi(t) : D \to \mathbb{R}_+$, with $D \subseteq \mathbb{R}_{++}$, is strictly convex and minimal at $t = 1$, with $\psi(1) = 0$. We call the univariante function $\psi(t)$ the *kernel function of the barrier function* $\Psi(v)$. Obviously,the resulting iteration bound will depend on the kernel function underlying

the algorithm, and our main tasks becomes to find a kernel function that minimizes the iteration bound.

All kernel functions considered so far are twice differentiable and go to infinity if either $t \searrow 0$ or $t \to \infty$. Thus they satisfy

1. $\psi(1) = \psi'(1) = 0$,

2. $\psi"(t) > 0$,

3. $\lim_{t\to 0} \psi(t) = \lim_{t\to\infty} \psi(t) = \infty$.

   The first and the second properties say tha $\psi(t)$ is a nonnegative strictly convex function such that $\psi(1) = 0$. Note that this implies that if $\psi(t)$ is twice differentiable, then it is completely determined by its second derivative:

$$\psi(t) = \int_1^t \int_1^\xi \psi"(\zeta) d\zeta d\xi. \tag{3.9}$$

   Moreover, the last property of the last definition expresses that $\psi(t)$ is coercive and has the barrier property. Having such a kernel function $\psi(t)$, its definition is extended to positive $n-dimensional$ vectors v by (3.8), thus yielding the induced (scaled) barrier function $\psi(v)$. In what follows we assume in this chapter that a kernel function satisfies the definition of kernel function.

   As we indicated, $\Psi(v)$ not only serves to define a search direction, but also acts as a measuriong of closeness of the current iterates to the $\mu - center$.In the analyses of the algorithm we also use the *norm-based proximity measure* $\delta(v)$ defined by

$$\delta(v) := \frac{1}{2} \|\nabla\Psi(v)\| = \frac{1}{2} \|d_x + d_y\|. \tag{3.10}$$

Note that since $\Psi(v)$ is strictly convex and minimal at $v = e$ we have

$$\Psi(v) = 0 \Longleftrightarrow \delta(v) = 0 \Longleftrightarrow v = e.$$

Thus the algorithm considred in this chapter uses the barrier function $\Psi(v)$ to measure closeness of the iterates to the $\mu-center$; as will become clear below, in the analysis of the algorithm $\delta(v)$ serves as a second proximity measure. Both measures are naturally determined by the kernel function.

4.5 Further conditions on the kernel function

In this sextion, we give the properties of the kernel function which are essential to our complexity analysis.

Lemma 36 *For $\psi(t)$, we have the following:*

1. *$\psi(t)$ is exponentially convex for all $t > 0$.*

2. *$\psi''(t)$ is monotonically decreasing for all $t > 0$.*

3. *$t\psi''(t) - \psi'(t) > 0$, for all $t > 0$.*

Proof. Using [[31], [33]] ■

Lemma 37 *For $\psi(t)$, we have the following:*

$$(m+1)(t-1)^2 \leq \psi(t) \leq \frac{1}{4(m+1)}\psi'(t)^2, \text{ for all } t > 0, \tag{3.11}$$

$$\psi(t) \leq \frac{(m+1)(m+2)}{2}(t-1)^2, \text{ for all } t > 0. \tag{3.12}$$

Now, let $\gamma : (0,\infty) \longrightarrow (1,\infty)$ be the inverse function of $\psi(t)$ for all $t \geq 1$, and $\rho : (0,\infty) \longrightarrow (0,1)$, be the inverse function of $-\frac{1}{2}\psi'(t)$ for all $t \in (0,1)$. Then we have the following lemma.

Lemma 38 *For $\psi(t)$, we have the following:*

$$\sqrt{\frac{s}{m+1}+1} \leq \gamma(s) \leq 1+\sqrt{\frac{s}{m+1}},\ s \geq 0, \tag{3.13}$$

and

$$\rho(s) \geq (\frac{m}{2s+m})^{\frac{1}{m+1}},\ s \geq 0. \tag{3.14}$$

Lemma 39 *Let $\gamma : (0,\infty) \longrightarrow (1,\infty)$ be the inverse function of $\psi(t)$ for all $t \geq 1$. Then we have*

$$\delta(\beta) \leq n\psi(\beta\gamma(\frac{\delta(v)}{n})),\ \beta \geq 1 \tag{3.15}$$

Proof. Using [3], we get the result. This completes the proof. ■

Lemma 40 *Let $0 \leq \theta \leq 1$, $v^+ = \frac{1}{\sqrt{1-\theta}}v$. If $\delta(v) \leq \tau$, then we have*

$$\delta(v^+) \leq n\frac{(m+1)(m+2)}{2(1-\theta)}(\sqrt{n}\theta+\sqrt{\frac{\tau}{m+1}})^2.$$

Proof. *Since $\frac{1}{\sqrt{1-\theta}} \geq 1$ and $\gamma(\frac{\delta(v)}{n}) \geq 1$, we have $\frac{\gamma(\frac{\delta(v)}{n})}{\sqrt{1-\theta}} \geq 1$. Using Lemma* 35,

with $\beta = \frac{1}{\sqrt{1-\theta}}$, (3.6), (3.7) *and* $\delta(v) \leq \tau$, *we have*

$$\begin{aligned}
\delta(v^+) &\leq n\psi(\frac{1}{\sqrt{1-\theta}}\gamma(\frac{\delta(v)}{n})), \\
&\leq n\frac{(m+1)(m+2)}{2}(\frac{1}{\sqrt{1-\theta}}\gamma(\frac{\delta(v)}{n})-1)^2, \\
&= n\frac{(m+1)(m+2)}{2(1-\theta)}(\gamma(\frac{\delta(v)}{n})-\sqrt{1-\theta})^2, \\
&\leq n\frac{(m+1)(m+2)}{2(1-\theta)}(1+\sqrt{\frac{\delta(v)}{(m+1)n}}-\sqrt{1-\theta})^2, \\
&\leq n\frac{(m+1)(m+2)}{2(1-\theta)}(\theta+\sqrt{\frac{\tau}{(m+1)n}})^2, \\
&\leq n\frac{(m+1)(m+2)}{2(1-\theta)}(\sqrt{n}\theta+\sqrt{\frac{\tau}{(m+1)n}})^2.
\end{aligned}$$

■

This completes the proof.

Denote

$$\Psi_0 = L(n,\theta,\tau) = n\frac{(m+1)(m+2)}{2(1-\theta)}(\sqrt{n}\theta+\sqrt{\frac{\tau}{(m+1)n}})^2, \tag{3.17}$$

then Ψ_0 is an upper bound for $\Psi(V)$ during the process of the algorithm.

4.6 Analysis of Algorithm

The aim of this chapter is to define a new kernel function and to obtain new complexity results for an (LCP) problem using the proximity function defined by the kernel function and following approach of *Bai et al. [3].* In the following, we compute a proper step size α and the decrease of the proximity function during an inner iteration and give the complexity results of the algorithm. For fixed $\mu > 0$.

4.6.1 Determining a default step size

Taking a step size α, we have new iterates

$$x^+ = x + \alpha\Delta x, \quad y^+ = y + \Delta y.$$

Let

$$x^+ = x(e + \alpha\frac{\Delta x}{x}) = x(e + \alpha\frac{d_x}{v}) = \frac{x}{v}(v + \alpha d_x),$$

$$y^+ = y(e + \alpha\frac{\Delta y}{y}) = y(e + \alpha\frac{d_y}{v}) = \frac{y}{v}(v + \alpha d_y).$$

So, we have

$$v^+ = ((v + \alpha d_x)^{\frac{1}{2}}(v + \alpha d_y))^{\frac{1}{2}}.$$

Since the proximity after one step is defined by

$$\delta(v^+) = \delta(((v + \alpha d_x)(v + \alpha d_y))^{\frac{1}{2}}).$$

By (i) in *Lemma* 36, we have

$$\delta(v^+) \leq \frac{1}{2}(\delta(v + \alpha d_x) + \delta(v + \alpha d_y)).$$

Define, for $\alpha > 0$

$$f(\alpha) = \delta(V^+) - \delta(V).$$

Therefore, we have $f(\alpha) \leq f_1(\alpha)$, where

$$f_1(\alpha) = \frac{1}{2}(\delta(v + \alpha d_x) + \delta(v + \alpha d_y)) - \delta(v). \tag{3.18}$$

Obviously,

$$f(0) = f_1(0) = 0.$$

Throughout this section, we assume that $\tau \geq 1$. Using *Lemma* 37 and the assumption that $\delta(v) \geq \tau$, we have $\delta(v) \leq \sqrt{m+1}$. By the definition of ρ, the largest step size of the worst case is given as follows:

$$\alpha^* = \frac{\rho(\delta) - \rho(2\delta)}{2\delta}. \tag{3.19}$$

Lemma 41 *Let the definition of ρ and α^* be as defined in* (3.19)*, then we have*

$$\alpha^* \geq \frac{1}{(m+1)(m+2)^{\frac{m+2}{m+1}}}.$$

Proof. Using [3], we get the result. ■

For using $\overline{\alpha}$ as the default step in the algorithm, define the $\overline{\alpha}$ as follows:

$$\overline{\alpha} = \frac{1}{3(m+1)(m+2)\delta^{\frac{m+2}{m+1}}} \tag{3.20}$$

4.6.2 Decrease of the proximity function during an inner iteration

Now, we show that our proximity function δ with our default step size $\overline{\alpha}$ is decreasing. It can be easily established by using the following result.

Lemma 42 *[36] Let $h(t)$ be a twice differentiable convex function with $h(0) = 0$, $h'(0) < 0$ and let $h(t)$ attains its(global) minimum at $t > 0$. If $h''(t)$ is creasing for $t \in [0, t^*]$, then*

$$h(t) = \frac{th'(0)}{2}.$$

Let the invariante function h be such that

$$h(0) = f_1(0) = 0,\ h'(0) = f_1'(0) = -2\delta^2,\ h''(\alpha) = f_2(\alpha) = 2\delta^2\Psi''(v - 2\alpha\delta).$$

Since $f_2(\alpha)$ holds the condition of the above lemma

$$f(\alpha) \leq f_1(\alpha) \leq f_2(\alpha) \leq \frac{f_2'(0)}{2}\alpha, \text{ for all } 0 \leq \alpha \leq \alpha^*.$$

We can obtain the upper bound for the decreasing value of the proximity in the inner iteration by the above lemma.

Theorem 43 *Let $\overline{\alpha}$ be a step size as defined in* (3.20) *and $\delta = \delta(v) \geq \tau = 1$. Then we have*

$$f(\overline{\alpha}) \leq -\frac{(m+1)^{\frac{-m-2}{2(m+1)}}}{3(m+2)}\Psi(V)^{\frac{m}{2(m+1)}}.$$

4.6.3 Iteration bound

We need to count how many inner iterations are required to return to the situation where $\delta(v) \leq \tau$ after a $\mu - update$. We denote the value of $\delta(v)$ after $\mu - update$ as δ_0 the subsequent values in the same other iterations are denoted as δ_k, $k = 1, ...$ If K denotes the total number of inner iterations in the outer iteration, then we have

$$\delta_0 \leq L = O(n, \theta, \tau),\ \delta_{K-1} > \tau,\ 0 \leq \delta_K \leq \tau.$$

And according to (3.15),

$$\delta_{k+1} \leq \delta_k - \frac{(m+1)^{\frac{-m-2}{2(m+1)}}}{3(m+2)}\delta_k^{\frac{m}{2(m+1)}}.$$

At this stage, we invoke the following lemma from [36].

Lemma 44 *Let t_0, $t_1, ..., t_k$ be sequence of positive numbers such that*

$$t_{k+1} \leq t_k - \beta t_k^{1-\nu}, \ k = 0, 1, ..., K-1,$$

where $\beta > 0$, $0 < \nu \leq 1$, then

$$K \leq \frac{t_0^{\nu}}{\beta\nu}.$$

Letting

$$t_k = \delta_k, \ \beta = \frac{(m+1)^{\frac{-m-2}{2(m+1)}}}{3(m+2)} \text{ and } \nu = \frac{m+2}{2(m+1)},$$

we can get the following lemma.

Lemma 45 *Let K be the total number of inner iterations in the outer iteration.*

Then we have

$$K \leq 6(m+1)^{\frac{3m+4}{2(m+1)}} \delta_0^{\frac{m+2}{2(m+1)}}.$$

Proof. Using *Lemma* 40, we get the result. ■

Now, we estimate the total number of iterations of our algorithm.

Theorem 46 *If $\tau \geq 1$, the total number of iterations is not more than*

$$6(m+1)^{\frac{3m+4}{2(m+1)}} \delta_0^{\frac{m+2}{2(m+1)}} \frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}.$$

Proof. In the algorithm, $n\mu \leq \varepsilon$, $\mu^k = (1-\theta)^k \mu^0$ and $\mu^0 = \frac{(x^0)^t y^0}{n}$. By a simple computation, we have

$$K \leq \frac{1}{\theta} \log \frac{n\mu^0}{\varepsilon}.$$

Therefore, the number of outer iterations is bounded above by $\frac{1}{\theta}\log\frac{n\mu^0}{\varepsilon}$. Multiplying the number of outer iterations by the number of inner iterations, we get an upper bound for the total number of iterations, namely,

$$6(m+1)^{\frac{3m+4}{2(m+1)}}\delta_0^{\frac{m+2}{2(m+1)}}\frac{1}{\theta}\log\frac{n\mu^0}{\varepsilon}.$$

■

Chapter 5

Numerical tests

In this section, we present two primal-dual small-step and large-step central path algorithms for solving a monoton linear complementarity program based on kernel function

$$\psi(t) = (m+1)t^2 - (m+2)t + \frac{1}{t^m}, t > 0, m > 4$$

The implementation is manipulated in DEV C++. Here we used *Iter* which means the iterations number produced by the algorithm. . Our tolerance is $\varepsilon = 10^{-6}$. For the update parameter, we have vary $0 < \theta < 1$.

5.0.4 Some examples

Let the following monoton linear complementarity problems, as follow:

Example 47 *The data for our first problem with* $n = 5$

$$M = \begin{pmatrix} 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ -2 & -1 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix}, \; q = \begin{pmatrix} -4 & -5 & 8 & 7 & 3 \end{pmatrix}$$

The following table summarises the results

Functions	Large update	Short update	θ	Iter
$\frac{t^2-1}{2} - \log t$	$O(n \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	84
			0.30	75
			0.60	35
			0.95	24
$\frac{t^2-1}{2} + \frac{t^{1-q}}{q(q-1)} - \frac{q-1}{q}(t-1), q > 1$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$	$O(q\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	83
			0.30	77
			0.60	64
			0.95	28
$\frac{t^2-1}{2} + \frac{(e-1)^2}{e(e^t-1)} - \frac{e-1}{e}(t-1)$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	79
			0.30	67
			0.60	56
			0.95	34
$\frac{1}{2}(t - \frac{1}{t})^2$	$O(n^{\frac{2}{3}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	82
			0.30	76
			0.60	45
			0.95	19
$\frac{t^{p+1}-1}{p+1} + \frac{t^{1-q}-1}{q-1}$, $p \in [0,1], q > 1$	$O(qn^{\frac{p+q}{q(1+p)}} \log \frac{n}{\varepsilon})$	$O(q^2\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	78
			0.30	75
			0.60	58
			0.95	27
$(m+1)t^2 - (m+2)t + \frac{1}{t^m}$, $t > 0, m > 4$	$O(m^{\frac{3m+1}{2m}} n^{\frac{m+1}{2m}} \log \frac{n}{\varepsilon})$	$O(m^{\frac{3m+1}{2m}} \sqrt{n} \log \frac{n}{\varepsilon})$	0.15	83
			0.30	63
			0.60	24
			0.95	12

Example 48 *Consider the following problem* $(\mathcal{MCQP})$ *with* $n = 10$

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 0.8 & 0.32 & 1.128 & 0.0512 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.32 & 0.128 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.32 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0512 & -1.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$q = \begin{pmatrix} -0.0256 & -0.064 & -0.16 & 5.59 & -1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The numerical results are presented in the following table.

Function	Large update	Short update	θ	Iter
$\frac{t^2-1}{2} - \log t$	$O(n \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	83
			0.30	77
			0.60	45
			0.95	14
$\frac{t^2-1}{2} + \frac{t^{1-q}}{q(q-1)} - \frac{q-1}{q}(t-1), q > 1$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$	$O(q\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	85
			0.30	78
			0.60	61
			0.95	23
$\frac{t^2-1}{2} + \frac{(e-1)^2}{e(e^t-1)} - \frac{e-1}{e}(t-1)$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	79
			0.30	61
			0.60	33
			0.95	23
$\frac{1}{2}(t - \frac{1}{t})^2$	$O(n^{\frac{2}{3}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	80
			0.30	78
			0.60	41
			0.95	17
$\frac{t^{p+1}-1}{p+1} + \frac{t^{1-q}-1}{q-1}$, $p \in [0,1]$, $q > 1$	$O(qn^{\frac{p+q}{q(1+p)}} \log \frac{n}{\varepsilon})$	$O(q^2\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	76
			0.30	75
			0.60	57
			0.95	17
$(m+1)t^2 - (m+2)t + \frac{1}{t^m}$, $t > 0, m > 4$	$O(m^{\frac{3m+1}{2m}} n^{\frac{m+1}{2m}} \log \frac{n}{\varepsilon})$	$O(m^{\frac{3m+1}{2m}} \sqrt{n} \log \frac{n}{\varepsilon})$	0.15	81
			0.30	43
			0.60	14
			0.95	5

Example 49 *Let $M \in \mathbb{R}^{10\times 10}$ et $q \in \mathbb{R}^{10}$ defined by*

$$M = \begin{pmatrix} 1 & 2 & 2 & . & . & . & 2 \\ 0 & 1 & 2 & . & . & . & 2 \\ 0 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & 2 \\ 0 & 0 & 0 & . & . & 0 & 1 \end{pmatrix}, \; q = \begin{pmatrix} -1 & . & . & . & -1 \end{pmatrix},$$

The numerical results are presented in the following table with $n = 10$

Function	Large update	Short update	θ	Iter
$\frac{t^2-1}{2}-\log t$	$O(n\log\frac{n}{\varepsilon})$	$O(\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	81
			0.30	72
			0.60	44
			0.95	11
$\frac{t^2-1}{2}+\frac{t^{1-q}}{q(q-1)}-\frac{q-1}{q}(t-1), q>1$	$O(qn^{\frac{q+1}{2q}}\log\frac{n}{\varepsilon})$	$O(q\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	80
			0.30	71
			0.60	61
			0.95	21
$\frac{t^2-1}{2}+\frac{(e-1)^2}{e(e^t-1)}-\frac{e-1}{e}(t-1)$	$O(n^{\frac{3}{4}}\log\frac{n}{\varepsilon})$	$O(\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	70
			0.30	51
			0.60	32
			0.95	12
$\frac{1}{2}(t-\frac{1}{t})^2$	$O(n^{\frac{2}{3}}\log\frac{n}{\varepsilon})$	$O(\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	81
			0.30	79
			0.60	40
			0.95	13
$\frac{t^{p+1}-1}{p+1}+\frac{t^{1-q}-1}{q-1}$, $p\in[0,1]$, $q>1$	$O(qn^{\frac{p+q}{q(1+p)}}\log\frac{n}{\varepsilon})$	$O(q^2\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	73
			0.30	74
			0.60	45
			0.95	27
$(m+1)t^2-(m+2)t+\frac{1}{t^m}$, $t>0, m>4$	$O(m^{\frac{3m+1}{2m}}n^{\frac{m+1}{2m}}\log\frac{n}{\varepsilon})$	$O(m^{\frac{3m+1}{2m}}\sqrt{n}\log\frac{n}{\varepsilon})$	0.15	63
			0.30	23
			0.60	11
			0.95	4

Example 50 *Take $n=15$, then the numerical results are presented in the following table.*

Function	Large update	Short update	θ	Iter
$\frac{t^2-1}{2} - \log t$	$O(n \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	60
			0.30	52
			0.60	24
			0.95	13
$\frac{t^2-1}{2} + \frac{t^{1-q}}{q(q-1)} - \frac{q-1}{q}(t-1)$	$O(qn^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon})$	$O(q\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	74
$, q > 1$			0.30	51
			0.60	44
			0.95	16
$\frac{t^2-1}{2} + \frac{(e-1)^2}{e(e^t-1)} - \frac{e-1}{e}(t-1)$	$O(n^{\frac{3}{4}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	63
			0.30	51
			0.60	32
			0.95	13
$\frac{1}{2}(t - \frac{1}{t})^2$	$O(n^{\frac{2}{3}} \log \frac{n}{\varepsilon})$	$O(\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	81
			0.30	79
			0.60	40
			0.95	13
$\frac{t^{p+1}-1}{p+1} + \frac{t^{1-q}-1}{q-1}$, $p \in [0,1]$	$O(qn^{\frac{p+q}{q(1+p)}} \log \frac{n}{\varepsilon})$	$O(q^2\sqrt{n} \log \frac{n}{\varepsilon})$	0.15	73
$, q > 1$			0.30	74
			0.60	45
			0.95	27
$(m+1)t^2 - (m+2)t + \frac{1}{t^m}$,	$O(m^{\frac{3m+1}{2m}} n^{\frac{m+1}{2m}} \log \frac{n}{\varepsilon})$	$O(m^{\frac{3m+1}{2m}} \sqrt{n} \log \frac{n}{\varepsilon})$	0.15	45
$t > 0, m > 4$			0.30	34
			0.60	9
			0.95	6

From the results obtained, the following remarks can be made:

Remark 51 *Our new kernel function produces a better execution time than the given kernel funcions.*

Remark 52 *In some cases, our new kernel function reduces the number of iterations compared to other kernel functions.*

Remark 53 *The number of iterations of the algorithm depends on the value of the parameter θ, in most cases, the larger θ gives a better number of iterations.*

Remark 54 *The results show very slow growth as n increases which is precisely what is expected for the $(\mathcal{IPM}s)$.*

5.0.5 Conclusion

We propose a new kernel function and primal-dual interior-point algorithms for to solve a monotone linear complementarity problems and analyze the complexity theoric and numeric based on central path method. We get for a large-update methods the number of iterations are $O(m^{\frac{3m+1}{2m}} n^{\frac{m+1}{2m}} \log \frac{(x^0)^T y^0}{\varepsilon})$ and $O(m^{\frac{3m+1}{2m}} \sqrt{n} \log \frac{(x^0)^T y^0}{\varepsilon})$ for small-update methods which are the best-known iteration bounds for such methods. Future research might focus on the extension to convex non-linear problem, convex quadratic semi definit problem. Finally, for the numerical tests, some strategies are used and indicate that our kernel function used in the algorithm is efficient.

Bibliography

[1] F. Alizadah, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM journal on Optimization, 1995.

[2] V. Alxeev, V. Tikhomirov, S. Fomine, Commande optimale, Traduction française Edition Mir (1982).

[3] Y. Q. Bai, M. El Ghami and C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, *SIAM J. Optim.* **15**(1) (2004) 101-128.

[4] S. Bazarra, H.D. Sherali and C.M. Shetty, " Nonlinear programming, theory and algorithms ", *Second edition*, 1993.

[5] R. Bellman, Introduction to matrix analysis, *Classics in Applied Mathematics,* Vol. **12** (SIAM, Philadelphia, 1995).

[6] D. Benterki, J. P. Crouzeix and B. Merikhi, A numerical feasible interior point method for linear semindefinite programs, *RAIRO Oper. Res.* 41 (2007) 49-59.

[7] G. B. Dantzig, Linear Programming and extensions, *Princeton University Press, Princeton,* N. J, 1963.

[8] E. Djeffal. L. Djeffal. D. Benterki, A procedure improving interior point algorithm for the linear programming, *24th Mini Euro Conference, June 23-26, Izmir, Turkey* (2010).

[9] E. Djeffal. L. Djeffal. D. Benterki, Extension of Karmarkar's algorithm for solving an optimization problem, *Advances in Applied Mathematics and Approximation Theor* (2013) 263–271.

[10] E. Djeffal. L. Mounia, A primal-dual interior-point method based on a new kernel function for linear complementarity problem, Asian-European Journal of Mathematics. Vol. **13**, No. **1** (2020) (16pages).

[11] A. V. Fiacco, G. P. Mccormick, Nonlinear Programming : Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968. Réimprimé par SIAM Publications, 1990.

[12] R. M. Freund, S. Mizuno, Interior Point Methods : *Current Status and Future Directions, Optima,* **51**, (October 1996)1-9.

[13] K. R. Frisch, The logarithmic potential method of convex programming, Technical Report, *University Institue of Economics, Oslo, Norway*, (1955).

[14] A. J. Goldman, A. W. Tucker, Theory of linear programming, in Linear Equalities and Related Systems, H. W. Kuhn and A. W. Tucker eds. *Princeton University Press,Princeton* N.J., 1956; pp:53 -97.

[15] M. Halicka, E. De Klerk, C. Roos, On the convergence of the central path in semidefinite optimization, *SIAM J. Optim.* **12** (2002) 1090-1099.

[16] P. Huard, Resolution of mathematical programming with nonlinear constraints by the method of centers, In J. Abadie, editor, *Nonlinear Programming*, North Holland, Amsterdam, The Netherlands, (1967) pages 207–219.

[17] N. Karmarkar, A new polynomial-time algorithm for linear programming the method for linearly constrained convex programming. *Combinatorica* **4** (1984) 373-395.

[18] A. Keraghel, Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar. Thèse en doctorat à l'université de Joseph Fourier- Grenoble I-.21 Octobre 2008.

[19] L. G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady*, **20** (1979); pp:191 -194.

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

[21] Z. Kebbiche, Etude et extensions d'algorithmes de points intérieurs pour la programmation non linéaire, thèse en doctorat, 2007.

[22] S. Kettab, D. Benterki, A relaxed logarithmic barrier method for semidefinite programming, *RAIRO. Oper. Res.* **49** (2015) 555-568.

[23] V. Klee, G. J. Minty, How good is the simplex algorithm, in Inequalities, *O. Shisha,ed., Academic Press, New York* (1972) 159 - 175.

[24] M. Kojima, S. Mizuno, A. Yoshise, A primal-dual interior point algorithm for linear programming, in : N. Megiddo (Ed.), Progress in Mathematical Programming : Interior Point and Related Methods, *Springer Verlag,* New York, (1989) 29-47.

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

[26] I. J. Lustig, A pratical approach to Karmarkar's algorithm, Technical report sol 85-5, Systems optimizatioin laboratory ; dep of operations res. STANFORD. univ ;Stanford California 94305 (1985).

[27] N. Megiddo, Pathways to the optimal set in linear programming, in : N. Megiddo (Ed.), Progress in Mathematical Programming : Interior Point and Related Methods, *Springer Verlag*, New York, 1989; pp:313-158.

[28] B. Merikhi, Etude comparative de l'extension de l'algorithme de Karmarkar et des méthodes simpliciales pour la programmation quadratique convexe, Thése de Magister, Institut de Mathématiques, University Ferhat Abbas, Sétif, Octobre 1994.

[29] R. D. C. Monteiro, I. Adler, M.G. C. Resende, A polynomial time primal dual affine scaling algorithm for linear and convex quadratic programming and its power series extension, *Math. Oper. Res. 15* (1990) 191-214.

[30] R. D. C. Monteiro, Primal-dual path-following algorithms for semidefinite programming, *SIAM J. Optim.* **7** (1997) 663-678.

[31] Y. E. Nestrov, A. S. Nemirovskii, Interior-point Polynomial Algorithms in Convex Programming, *SIAM Stadies in Applied and Numerical Mathematics,* Vol. **13** (SIAM, Philadelphia, 1994).

[32] M.Padberg, A different convergence proof of the projective method for linear programming, Operations research letters, (1985), Volume **4**, number **6**.

[33] J. Peng, C. Roos, T. Terlaky, A New Paradigm for Primal-dual Interior-Point Methods,Princeton University Press, Princeton, NY, 2002.

[34] J. Peng, C. Roos and T. Terlaky, Self-Regularity: A new Paradigm for primal-dual interior-point algorithms (*Princeton University Press, Princeton, NJ, 2002).*

[35] J. Peng, C. Roos and T. Terlaky, Self-Regularity: Functions and new search directions for linear and semidefinite optimization, *Math. Program.* **93** (2002) 129-171.

[36] Y. Qian, A polynomial predictor–corrector interior point algorithm for convex quadratic programming, Acta Mathematica Scientia, 26(2) (2006), 265–279.

[37] M. Reza Peyghami, S.Fathi Hafshejani, An interior point algorithm for solving convex quadratic semidefinite optimization problems using a new kernel function, *Irnian Journal of Mathematical Sciences and Informatics.* Vol **12**, No 1 (2017) 131-152.

[38] C. Roos, T. Terlaky, J.Ph. Vial, Theory and algorithms for linear optimization, in An Interior Approach, John Wiley and Sons, Chichester, UK, 1997.

[39] N. Z. Shor, Utilization of the operation of space dilatation in the minimization of

convex functions, Kibernetika, 1(1970); pp:6-12. Traduction anglaise : Cybernetics,6; pp:7-15.

[40] M. J Todd, B.P. Burell, An extension of Karmarkar's algorithm for linear programming using dual variables, *Algorithmica* **1** (1986) 409–424.

[41] L. Vandenberghe, S. Boyd, Primal-dual potential reduction method for problems involving matrix inequalities, *Math. Programming. Series B.* ***69*** (1995) 205-236.

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

[43] Y.Ye, A class of projective transformations for linear programming, SIAM J. Comput. Vol 19. No. 3, pp 457-466, June 1990.

[44] Y Ye, Interior Point Algorithms: Theory and Analysis, Wiley-Interscience, 1997.

[45] Y. Zhang, On the convergence of a class of infeasible-point methods for the horizontal linear complementarity problem, *SIAM J. Optim.* **4** (1994) 208-227.

ملخص:
من المعروف جيدًا أن أساليب النقاط الداخلية هي الأكثر فاعلية في حل مسائل البرمجة المثالية. هذه الطرق تمتاز بتقاربها الحدودي وسلوكها العددي الجيد لايجاد مجموعة الحلول. في هذا البحث، كان اهتمامنا حول الدراسة النظرية، والعددية، والخوارزمية لطرق النقطة الداخلية لمشكلة التكامل الخطي.
في الواقع نهتم بطريقة المسار المركزي عبر دالة النواة، وقد قمنا باقترح دالة نواة جديدة التي أعطت لنا أفضل نتائج من حيث تكلفة الخوارزمي.

الكلمات المفتاحية: مشكلة تربيعية محدبة، طريقة النقطة الداخلية،وظيفة النواة، تكلفة الخوارزمية.

Abstract :

It is well that interior point methods are the most efficient to solve an optimization problems. These methods are characterized by their polynomial convergence to find a set of solutions. In this research, we are interested a theoretical, numerical and an algorithmic study of interior-point methods for linear complementarity problem.

Indeed, we are interested in a central trajectory method via a kernel function, we proposed new kernel function that give the best-known complexity results.

Keywords: Convexe quadratic problem, Interior point method, Kernel function, Algorithmic complexity.

Résumé :

Il est bien connu que les méthodes de points intérieurs sont les plus efficaces pour résoudre les problèmes d'optimisation. Ces méthodes se caractérisent par leur convergence polynomiale à la frontière et leur bon comportement numérique pour trouver la solution. Dans cette recherche, nous nous intéressons à une étude théorique, numérique et algorithmique des méthodes de points intérieurs pour le problème de complémentarité linéaire.

En effet, nous nous intéressons à une méthode de trajectoire centrale via une fonction noyau, nous avons proposé de nouvelles fonctions noyau qui donnent les résultats de complexité les plus connus.

Mots clés : Problème quadratique convexe, Méthode des points intérieur, Fonction noyau, Complexité algorithmique.