REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE Ferhat ABBAS SETIF 1

THESE

Présentée

Pour l'obtention du diplôme de DOCTORAT LMD 3ème cycle

Filière :Mathématiques

Spécialité : Optimisation et Contrôle

Par

Mme Nadia HAZZAM

THEME

Méthodes de points intérieurs appliquées au problème de complémentarité linéaire

Soutenue le :30/09/2021

Devant le jury :

ROUMILI Hayet	**Professeur**	**Univ. Ferhat ABBAS Sétif 1**	**Président**
KEBBICHE Zakia	**Professeur**	**Univ. Ferhat ABBAS Sétif 1**	**Rapporteur**
BOUAFIA Mousaab	**MCA**	**Univ. 8 mai 1945 Guelma**	**Examinateur**
GRAR Hassina	**MCA**	**Univ. Ferhat ABBAS Sétif 1**	**Examinateur**

Acknowledgements

I would like to express the deepest appreciation to my director of thesis **Prof. Kebbiche Zakia** for her guidance and constant supervision as well as for providing necessary information regarding this research and for her support in completing this work.

I would also like to thank the commitee members, **Roumili Hayet** professor in the university of Ferhat Abbas Setif 1, **Bouafia Mousaab** lecturer in the university of 8 mai 1945 Guelma and **Grar Hassina** lecturer in the university of Ferhat Abbas Setif 1.

I would like to express my gratitude to **Prof. Achache Mohamed**, **Dr. Addoune Smail** for their help, advice and motivation.

Also I express my thanks to my friends; **Dr. Hechaichi Hadjer, Dr. Zerari Amina** and **Dr. Bouras Ikram** for their support and help to complete my thesis. Last, I am extremely grateful to my family for their love, prayers, caring and sacrifices for educating.

Abstract

The systematic study of complementarity problems began with the work of R.W Cottle in the 1960s. The importance of complementarity can be measured by the crucial role it plays in solving many problems in different fields. These problems first manifested themselves in the optimality conditions of the optimization problems; the Karush, Kuhn and Tucker conditions of a linear or a quadratic program are equivalent to a linear complementarity problem. This thesis interests with the analysis and implementation of interior-point methods for solving horizontal linear complementarity problem. In chapter 1, we give some notations and definitions that will be used in the following chapters and we introduce the mathematical reformulation of the linear complementarity problem and some of its applications. In chapter 2, we present a theoretical and practical study of the transformation of an absolute value equation to an horizontal linear complementarity problem by introducing an infeasible primal-dual central path method. In chapter 3, we propose a feasible primal-dual interior-point methods for horizontal linear complementarity problem. The method is based on a new class of parametric kernel functions. We show that the corresponding algorithm has the best known iteration bound for large-update methods. Then, we illustrate the performance of the proposed kernel function by some comparative numerical results. In chapter 4, a new variant of Mehrotra type-predictor-corrector algorithm is proposed for horizontal linear complementarity problem. We demonstrate the theoretical efficiency of this algorithm by showing its polynomial complexity. We test the practical efficiency and the validity of our algorithm by running some computational tests. Finally, this algorithm is compared with a Mehrotra-type predictor-corrector algorithm [63].

Keywords: Horizontal linear complementarity problem, interior-point method, kernel function, predictor-corrector algorithm, Mehrotra's algorithm, complexity bound.

Contents

Introduction

The aim of this doctoral thesis is the complexity analysis and numerical implementation of interior-point methods for $P_*(\kappa)$-horizontal linear complementarity problem ($P_*(\kappa)$-HLCP). The name of this problem has undergone several changes. It has been called the "composite problem", the "fundamental problem" and the "complementary pivot problem". In 1965, the current name "linear complementarity problem (LCP)" was proposed by Cottle. It was later used in a paper by Cottle, Habetler and Lemke (1970). Probably the earliest publication containing an explicitly stated LCP is one by Du Val (1940). This paper, part of the literature of algebraic geometry, used a problem of the form (q, M) to find the least element (in the vector sense) of the linear inequality system $q + Mz \geq 0, z \geq 0$. Ordinarily, problems of this sort have no solution, but when the matrix M has special properties, a solution exists and is unique. The linear complementarity problem was conceived as a unifying formulation for the linear and quadratic programming problems as well as for the bimatrix game problem which includes the standard linear complementarity problem, linear and quadratic optimization problems (LO and QO). In fact, quadratic programs have always been—and continue to be—an extremely important source of applications for the linear complementarity problem. Several highly effective algorithms for solving quadratic programs are based on the linear complementarity problem formulation. As far as the bimatrix game problem is concerned, the linear complementarity problem formulation was instrumental in the discovery of a super constructive tool for the computation of an equilibrium point. After the landmark paper of Karmarkar [34] for linear problem (LP), the interior-point methods (IPMs) became an active area of research. Nowadays, introducing an efficient variant of the polynomial time interior-point methods with low complexity results is the main challenge in this area of research. Among the different variants of interior-point methods, the most efficient methods are the so-called primal-dual path following methods from a computational point of view. The primal-dual central path was initially described by Megiddo [43], and the evolution of primal-dual path following methods was surveyed in the paper by Gonzaga [26], followed by the algorithm of Kojima, Mizuno, and Yoshise [38]. Because the linear complementarity problems have a close connection with linear problem, so that many algorithms such that interior-point algorithms

were extended to linear complementarity problems without losing their attributes of simplicity and good practical behavior [58]. Kojima et al. [37] extended all previously results for linear problem to a linear complementarity problem with a positive semi-definite matrix. Since then, many interior point algorithms have been prolonged to $P_*(\kappa)$-LCP. In chapter 2, we present a theoretical and practical study of the transformation of an absolute value equation to a horizontal linear complementarity problem by introducing an infeasible primal-dual central path method.

It's well known that there is a gap between the practical behaviour of IPM and the theoretical performance results especially for the so-called large-update methods. Much research work has been done on IPM theoretically and practically. Recently, a particularly interesting alternative is being studied, that of IPMs based on barrier functions that are defined by a large class of univariate functions called kernel functions that have been successfully used to design new IPMs for linear optimization. Kernel functions play an important role in the analysis of primal-dual interior-point algorithms. They are not only used to determine search directions but also to measure the distance between the given iteration and the μ-center of those algorithms. Peng et al. [48] introduced a primal-dual IPM for LO based on a new class of barrier function that is defined by the so-called self-regular kernel function. They significantly improved the theoretical complexity and obtained the best known theoretical iteration bound for LO with large-update primal-dual IPM, namely $O\left(\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$. Recently, Bai et al. [11] defined the concept of eligible kernel functions which require some conditions and proposed primal-dual IPMs for LO problems based on these functions, and some of these methods achieved the best known complexity results for both large and small-update methods.

Cho [18] and Cho et al. [17] extended these algorithms from LO to $P_*(\kappa)$-LCPs and obtained the similar complexity results as LO problems for large-update methods. Wang et al. [57] generalized polynomial IPMs for LO problem to $P_*(\kappa)$-HLCP based on a finite kernel function, which was first defined in [5], and obtained the same iteration bounds for large and small-update methods as in LO problem. Lee et al. [39] proposed new classes of kernel functions whose barrier term is exponential power of exponential functions for $P_*(\kappa)$-HLCP. Asadi et al. [8] proposed an infeasible interior point algorithm for $P_*(\kappa)$-HLCP based on a kernel function. Fathi-Hafshejani et al. [27] presented an IPM for $P_*(\kappa)$-HLCP based on a new proximity function. El Ghami et al. [23] extended IPMs for LO problems to the $P_*(\kappa)$-LCPs based on eligible kernel functions, which were defined in [11], and proposed large as well as small-update methods. Bouafia et al. [14] proposed an IPM for LO problems based on a kernel function whose barrier term is a trigonometric function. El Ghami et al. [24] and Fathi-Hafshejani et al. [28] proposed a new trigonometric kernel function for $P_*(\kappa)$-LCP. Ji et al. [33] introduced a primal-dual large-update interior point algorithm for $P_*(\kappa)$-HLCP based on a new class of kernel function.

In an attempt to improve theoretical complexity of large-update methods, in chapter 3, we introduce a primal-dual interior-point algorithm for $P_*(\kappa)$-HLCP based on a new family of kernel functions with trigonometric barrier term [29]. We provide the polynomial iteration complexity. We examine the performance of our algorithm by comparing its implementation results with other recent kernel functions.

Among various variants of interior-point methods, the so-called predictor-corrector methods have attracted much attention in the interior method community due to the high efficiency and have been widely used in several interior-point method based optimization packages [21, 31, 32, 44, 46, 49, 54, 62]. Mehrotra's predictor-corrector algorithm for LP was proposed in 1989 in which the coefficient matrices in both predictor step and corrector step are the same and it needs less computational efforts than other methods so the performance of practical primal-dual algorithms is significantly improved. After that, many variants of this algorithm have been studied and discussed. Illés et al. [31], Miao [45] analyzed a version of the Mizuno-Todd-Ye predictor-corrector interior-point algorithm for $P_*(\kappa)$-LCP. In 2012, Liu et al. [41] proposed two Mehrotra-type predictor-corrector algorithms for sufficient LCP. Jarre et al. [32] proposed an extension of Mehrorta's predictor-corrector algorithm and gave a modification to allow some repetition of the corrector step. Salahi et al. [55] discussed the polynomial complexity of a version of Mehrotra's predictor-corrector algorithm and gave a numerical example showing that the choice of some terms in this algorithm may give a small steps, so that, they introduce a safeguard strategy to avoid these small steps and also the poorly performance of the affine scaling direction. Yang et al. [59] and Liu et al. [42] study new infeasible versions of Mehrotra-type algorithm. Recently, Asadi et al. [6], proposed a Mehrotra-type predictor-corrector algorithm for LP which is a variant of the algorithm presented in [9]. The analysis leads to an improvement of factor n^3 in the complexity analysis, which is a significant improvement in the theoretical complexity of the algorithm given in [9]. In 2019, Zhou et al. [63] proposed a variant of Mehrotra type predictor-corrector algorithm in which a safeguard step is used to avoid small step sizes. They proved the polynomial iteration complexity of the new algorithm.

Motivated by these works, in chapter 4, we propose a new Mehrotra type interior-point algorithm to $P_*(\kappa)$-HLCP. In this algorithm, we use the negative infinity norm neighborhood and the corrector search direction is obtained by solving the same system given in [63] but with a different centering parameter. We demonstrate the theoretical efficiency of this algorithm by showing its polynomial complexity, then we examine its performance by solving some problems. Finally, we compare our algorithm with the Mehrotra-type predictor-corrector algorithm presented in [63] by solving some problems via both algorithms.

The thesis is organized as follows:

In chapter 1: we give fundamental definitions and notations that will be used throughout the thesis. Also, we present some applications of the linear complementarity problem.

In chapter 2: we present an application of the horizontal linear complementarity problem by introducing a theoretical and numerical study of an infeasible primal-dual central path-following method.

In chapter 3: we propose a primal-dual interior-point method based on a new class of kernel functions to solve horizontal linear complementarity problem. The polynomial iteration complexity is provided. Some illustrative and comparative numerical results are reported.

In chapter 4: a new variant of Mehrotra type predictor-corrector algorithm is proposed to solve horizontal linear complementarity problem. We demonstrate the theoretical efficiency of this algorithm by showing its polynomial complexity. We test the practical efficiency and the validity of our algorithm by running some computational tests. Finally, this algorithm is compared with a Mehrotra-type predictor-corrector algorithm [63].

These chapters are followed by a general conclusion and bibliography.

Chapter 1

Fundamental definitions and notations

Our purpose in this chapter is to collect the essential background materials needed for the rest of the thesis.

1.1 Notations

$\mathbb{R}^n$	:	the real n-dimensional space;
$\mathbb{R}^n_+$	:	the nonnegative orthant of $\mathbb{R}^n$;
$\mathbb{R}^n_{++}$	:	the positive orthant of $\mathbb{R}^n$;
$\mathbb{R}^{n\times n}$	:	the set of all $n \times n$ squared matrices;
x^T	:	the transpose of vector x;
e	$=$	$(1, ..., 1)^T$; the n-dimensional vector of ones;
xy	$=$	$(x_1y_1, ..., x_ny_n)^T$; the Hadamart product;
x^Ty	$=$	$\sum_{i=1}^{n} x_iy_i$;
$\langle x, y\rangle$	$=$	x^Ty
$\frac{x}{y}$	$=$	$\left(\frac{x_1}{y_1}, ..., \frac{x_n}{y_n}\right)^T$, $y_i \neq 0$ for all $i = 1, \ldots, n$;
$\sqrt{x}$	$=$	$\left(\sqrt{x_1}, ..., \sqrt{x_n}\right)^T$, $x \geq 0$;
x^{-1}	$=$	$\left(\frac{1}{x_1}, ..., \frac{1}{x_n}\right)^T$, $x_i \neq 0$ for all $i = 1, \ldots, n$;
$\lvert x\rvert$	$=$	$(\lvert x_1\rvert, ..., \lvert x_n\rvert)^T$; the componentwise absolute value of a vector $x \in \mathbb{R}^n$
$\lVert x\rVert$	$=$	$\sqrt{\sum_{i=1}^{n} x_i^2}$; the 2-norm of a vector x;
$\lVert x\rVert_1$	$=$	$\sum_{i=1}^{n} \lvert x_i\rvert$; the 1-norm of a vector x;
$\lVert x\rVert_\infty$	$=$	$\max(\lvert x_1\rvert, ..., \lvert x_n\rvert)$; the infinite norm of vector x

$diag\,(x)$	:	the diagonal matrix where the diagonal elements are the components of x i.e x_i;
I	:	the identity matrix
$\sigma_{min}\,(A)\,(\sigma_{max}\,(A))$	:	the minimal (maximal) singular value of the matrix A;
$\lambda_{min}\,(A)\,(\lambda_{max}\,(A))$	:	the minimal (maximal) eigenvalue of the matrix A.
$min(a, b)$	:	the componentwise minimum of two vectors a and b.
$max(a, b)$	:	the componentwise maximum of two vectors a and b.

1.2 Preliminaries

1.2.1 Definitions

Definition 1.2.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is **differentiable** at $a \in \mathbb{R}^n$ if and only if there exists a linear function $L : \mathbb{R}^n \to \mathbb{R}$ such that*

$$f(x) = f(a) + L(x-a) + o(x)\|x-a\|, \lim_{x\to a} o(x) = 0.$$

If this is the case, then $L = df(a)$ is the differential of f at the point a.

Definition 1.2.2. *A set $D \subset \mathbb{R}^n$ is called **convex** if*

$$tx + (1-t)y \in D, \quad \text{for all } x, y \in D \text{ and } t \in [0,1].$$

- *Given a convex set D, a function $f : D \to \mathbb{R}$ is called **convex** if*

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \tag{1.1}$$

  *for all $x, y \in D$ and $t \in [0,1]$, f is called **strictly convex** if the inequality (1.1) is strict for all $x, y \in D, x \neq y$ and $t \in]0,1[$.*

- *A function f is **concave** over a convex set if and only if the function $-f$ is a convex function over this set.*

Definition 1.2.3. *A barrier function of a set D is a continuous function which approaches infinity as the boundary of D is approached from the interior.*

Definition 1.2.4. *A function $\psi : \mathbb{R}_{++} \to \mathbb{R}_+$ is called a **kernel function** if ψ is twice differentiable and satisfies the following conditions*

$$\psi'(1) = \psi(1) = 0, \quad \psi''(t) > 0, \text{ for all } t > 0, \text{ and } \lim_{t\to 0^+} \psi(t) = \lim_{t\to\infty} \psi(t) = \infty.$$

Note that the two first conditions mean that f is a strictly convex function and allow a minimum at $t = 1$ and the last one creates a barrier at the boundary of the definition set, so this function has also the name "barrier function".

1.2.2 Newton's method for nonlinear system

Most iterative algorithms in optimization (for example interior-point methods) have two basic ingredients, a procedure for determining the step and measure of the desirability of each point in the search space, the search direction procedure has its origins in Newton's method for the nonlinear equations. So we give the following definitions.

Given a system of nonlinear equations

$$f(x) = \begin{bmatrix} f_1(x_1, ..., x_n) \\ f_2(x_1, ..., x_n) \\ \vdots \\ f_n(x_1, ..., x_n) \end{bmatrix} = 0$$

with $x = (x_1, ..., x_n)^T \in \mathbb{R}^n$ and $f_i, i = 1, ..., n$, are real functions.

- The Jacobian matrix $J(x)$ of f is as follows

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \cdots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \frac{\partial f_n}{\partial x_2}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix}$$

  where $\frac{\partial f_i}{\partial x_j}(x)$ is the j^{th} partial derivative of f_i.

- The Hessian matrix is a matrix of second order partial derivatives, define as follows

$$H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{bmatrix}.$$

Newton's method

Given a vector x^0 representing the first guess for the solution, successive approximations to the solution are obtained from

$$x^{k+1} = x^k - J^{-1}(x^k) f\left(x^k\right) = x^k + \Delta x^k,$$

where Δx^k is the Newton's direction, solution of the system

$$J(x^k)\Delta x^k = -f(x^k).$$

1.3 Linear complementarity problem (LCP)

1.3.1 Problem statement

The linear complementarity problem (LCP) consist in finding two vectors (x, y) in $\mathbb{R}^n \times \mathbb{R}^n$ such that

$$y = Mx + q, \quad x^T y = 0, \quad x \geq 0, \quad y \geq 0, \tag{1.2}$$

where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ are given, or to show that no such a solution exists.
In the so-called optimality conditions the first constraint represents feasibility, whereas the last equation is the so-called complementary condition. The nonnegativity constraints make the problem nontrivial: only iterative methods can find a solution of a linear system involving inequality constraints. The complementary condition is nonlinear, which makes the resolution of the system (1.2) extra hard.
We denote

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : y = Mx + q, x \geq 0, y \geq 0\}$$

the set of feasible solutions of (1.2), and

$$\mathcal{F}^0 = \{(x, y) \in \mathcal{F} : x > 0, y > 0\}$$

the set of its strict feasible solutions.

Classes of matrices

LCPs are typically classified according to the classes of the matrix M. In this regard, the following definitions are given:

- A matrix $M \in \mathbb{R}^{n\times n}$ is called **positive semidefinite (definite)** if for every $x \in \mathbb{R}^n$ we have

$$x^T Mx \geq 0 \quad (x^T Mx > 0, x \neq 0).$$

- A matrix $M \in \mathbb{R}^{n\times n}$ is called a P**-matrix (**P_0**-matrix)** if all of its principal minors are positive (nonnegative), i.e, for all $x \neq 0$ there exist $i \in \{1, \ldots, n\} : x_i(Mx)_i > 0, \quad (x_i(Mx)_i \geq 0)$.

- Kojima et al. [37] have introduced a class of matrices called $P_*(\kappa)$-matrix, we say that a matrix $M \in \mathbb{R}^{n\times n}$ is a $P_*(\kappa)$-matrix if

$$(1+4\kappa)\sum_{i\in I^+(x)} x_i\left(Mx\right)_i + \sum_{i\in I^-(x)} x_i\left(Mx\right)_i \geq 0, \text{ for all } x \in \mathbb{R}^n,$$

where κ is a nonnegative real and

$$I^+\left(x\right) = \{i : x_i\left(Mx\right)_i \geq 0\}, I^-\left(x\right) = \{i : x_i\left(Mx\right)_i < 0\}.$$

We may define the set of P_*-matrices as the union of $P_*(\kappa)$-matrices in the following way

$$P_* = \bigcup_{\kappa\geq 0} P_*(\kappa).$$

These properties are closely related to the existence and uniqueness of solutions of LCPs. The following results was proved by Cottle et al. [20].

Theorem 1.3.1. *(Theorem 3.1.2, [20]) Let M be a positive semidefinite matrix, if the corresponding LCP is feasible then it is solvable.*

Theorem 1.3.2. *(Theorem 3.3.7, [20]) a matrix $M \in \mathbb{R}^{n\times n}$ is a P-matrix if and only if the corresponding LCP has a unique solution for all vectors $q \in \mathbb{R}^n$.*

LCP and optimization problems

The linear complementarity plays a very important role in solving several problems in different domains: linear problem, convex quadratic programing, variational inequations, mechanic...
The LCP (1.2) generalises the linear problem and the convex quadratic problem as follows:

- **Linear programming**

  Consider the linear program (LP)

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

  where $A \in \mathbb{R}^{m\times n}$ is of full rank , $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

  This problem is convex and the constraints are qualified, then the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient, so they are equivalent to an LCP as follows: $x \in \mathbb{R}^n_+$ is

an optimal solution of the previous LP if and only if there exists $y \in \mathbb{R}^m_+$ and $\lambda \in \mathbb{R}^n_+$ such that the KKT conditions are written as an LCP

$$\begin{cases} \text{Find } (w,z) \in \mathbb{R}^{n+m} \times \mathbb{R}^{n+m} \text{ such that:} \\ w = Mz + q \\ w^T z = 0, \\ (w,z) \geq 0, \end{cases}$$

where

$$M = \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ b \end{bmatrix}, \quad w = \begin{bmatrix} \lambda \\ v \end{bmatrix}, \quad v = b - Ax \text{ and } z = \begin{bmatrix} x \\ y \end{bmatrix}.$$

- **Quadratic programming**

  1/ Consider the quadratic problem

$$\begin{cases} \min \quad \frac{1}{2} x^T Q x + c^T x \\ Ax \leq b \\ x \geq 0, \end{cases}$$

  where $Q \in \mathbb{R}^{n \times n}$ is a symmetric semidefinite matrix, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.

  This problem is convex and the constraints are qualified, which we denoted by (CQP), then the KKT conditions are necessary and sufficient, x is an optimal solution of CQP if and only if there exists $y \in \mathbb{R}^m_+$ and $\lambda \in \mathbb{R}^n_+$ such that the KKT conditions can be written as an LCP

$$\begin{cases} \text{Find } (w,z) \in \mathbb{R}^{n+m} \times \mathbb{R}^{n+m} \\ w = Mz + q, \\ w^T z = 0, \\ (z,w) \geq 0, \end{cases}$$

  where

$$M = \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ b \end{bmatrix}, \quad z = \begin{bmatrix} x \\ y \end{bmatrix}, \quad w = \begin{bmatrix} \lambda \\ v \end{bmatrix} \text{ and } v = b - Ax.$$

2/ Consider the LCP

$$\begin{cases} \text{Find } x \in \mathbb{R}^n \\ Mx + q \geq 0, x \geq 0, \\ x^T(Mx + q) = 0, \end{cases}$$

* If M is symmetric, the LCP constitutes the KKT optimality conditions of quadratic programs with simple nonnegativity constraints of the variables that is:

$$\begin{cases} min \quad \frac{1}{2}x^T(Mx + q) \\ x \geq 0. \end{cases}$$

* If M is asymmetric, in this case we can associate with the LCP the following quadratic program

$$\begin{cases} min \quad x^T(Mx + q) \\ Mx + q \geq 0, \\ x \geq 0. \end{cases}$$

Notice that the objective function of this quadratic program is always bounded bellow by zero on the feasible set. It is trivial to see that a vector x is a solution of the LCP if and only if it is a global minimum of this quadratic program with an objective value of zero. In the study of LCP, one normally does not assume that the matrix M is symmetric. The formulation of this quadratic program is useful in that it allow one to specialize the results from quadratic programming theory to the general LCP.

1.3.2 Resolution methods

Many algorithms have been developed for solving the linear complementarity problem, the existing methods are extensions of methods designed for linear programming.

Pivotal methods

The first class of methods to be developed for the LCPs are the pivotal methods or simplicial methods. These are variants of the simplex method in which the concept of pivot is used. These algorithms require the class of positive semidefinite matrices and the P-matrices, this class of LCPs is equivalent to the class of convex quadratic programs. We cite

- Principal pivoting methods
- Lemke's method.

The simplicial methods start at a solution of an artificially set up simple system, and trace a path through the n-dimensional simplices of the triangulation, which, when the method works, terminates with a simplex that contains an easily computed approximate solution of the original system. Using a homotopy interpretation of this path, a variety of other homotopy and path tracing algorithms have been developed for solving systems of nonlinear equations. These algorithms are known by their good numerical behavior, while in theory they are considered inefficient because of their exponential complexity expressed by the total number of possible pivots.

Interior-point methods

The other important class of methods for the LCPs are the interior-point methods. Those approaches are proposed by Kojima et al. originated from the algorithm introduced by Karmarkar in 1984 for solving linear programs. The most successful interior-point methods follow a path in strict feasible set of the problem (hence the name interior-point methods) in an effort to reduce $x^T y$ to zero. They generally demonstrate an excellent practical and theoretical behavior (polynomial complexity). We cite

- Projective methods,
- Affine-scaling methods,
- Potential-reduction methods,
- Path-following methods.

We focus a little on the last category of the interior-point methods which is the objective of this thesis, in particular, on the central path-following methods.
For each of these classes of methods, it is possible to define primal, dual and primal-dual variants. We can classify the interior-point methods depending on the type of algorithms they employ. For this purpose, we have: small-update algorithms, large-update algorithms and predictor-corrector algorithms.

Central path-following methods for LCP

Consider the linear complementarity problem

$$\begin{cases} \text{Find } (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \text{ such that} \\ y = Mx + q, \\ x^T y = 0, x \geq 0, y \geq 0, \end{cases} \tag{1.3}$$

where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ are given, and

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : y = Mx + q, x \geq 0, y \geq 0\},$$

$$\mathcal{F}^0 = \{(x, y) \in \mathcal{F} : x > 0, y > 0\}.$$

In what follows, the matrix M is assumed to be positive semidefinite (in this case the LCP is called monotone). We further assume that the previous LCP has a strictly feasible vector.
The LCP is equivalent to the following quadratic program

$$\begin{cases} \min\limits_{(x,y)} x^T y \\ y = Mx + q, \\ (x, y) \geq 0, \end{cases} \tag{1.4}$$

in the sense that if (x^*, y^*) is a solution of (1.4) with $x^{*T} y^* = 0$ then (x^*, y^*) is a solution of (1.3).
The concept of a central path can be introduced using a logarithmic barrier function, we associate with (1.4) the following barrier problem

$$\begin{cases} \min\limits_{(x,y)} f_\mu(x, y) = x^T y - \mu \sum\limits_{i=1}^{n} \log x_i y_i, \\ y = Mx + q, \\ (x, y) > 0, \end{cases} \tag{1.5}$$

where $\mu > 0$, if $\mu \to 0$, the solution of (1.5) converges to the solution of (1.3), so we solve a series of problems (1.5) by decreasing the values of μ until we obtain a solution of (1.3). It's well known that the function f_μ is strictly convex (Proposition 3 in [35]).

Theorem 1.3.3. *(Theorem 4 in [35]) If M is positive semidefinite and $\mathcal{F}^0 \neq \emptyset$, the (1.5) have a unique solution for all $\mu > 0$.*

Theorem 1.3.4. *(Theorem 5 in [35]) Let $\mu > 0$ and M be a positive semidefinite matrix, (x, y) is a solution of (1.3) if and only if (x, y) satisfies the following system:*

$$\begin{cases} XYe - \mu e = 0, \\ y = Mx + q, \\ (x, y) > 0. \end{cases} \tag{1.6}$$

Therefore, solving (1.5) is equivalent to solving the system (1.6).

Kojima et al. in [37] showed that if an LCP is strictly feasible then the central path of the underlying LCP is well-defined. Let $(x(\mu), y(\mu))$ denote the unique solution of the nonlinear system (1.6) for a given $\mu > 0$. The set $T = \{(x(\mu), y(\mu)) : \mu > 0\}$ is called the central path.
Since the first equation of the problem (1.6) is nonlinear, a direct resolution is generally difficult. So, we look for an approximate solution $(x(\mu), y(\mu))$ to the following system

$$F_\mu(x, y) = 0, \tag{1.7}$$

where $(x, y, \mu) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$ and $F_\mu : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ is defined by

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

(x, y) is a solution of the LCP if and only if it is a solution of (1.7). Apply Newton's method to the system (1.7) from a strict feasible point (x, y), we obtain

$$\nabla F_\mu(x, y) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -F(x, y)$$

i.e.,

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

The new iterate is

$$\left(x^+, y^+\right) = (x, y) + (\Delta x, \Delta y).$$

Proximity measure

The quality of each found solution is measured by a so-called proximity measure

$$\delta(x, y, \mu) = \min_{\mu > 0} \|XYe - \mu e\| = \left\| XYe - \frac{x^T y}{n} e \right\|.$$

$(x, y) \in \mathcal{F}^0$ is a point of the central path if

$$\left\| XYe - \frac{x^T y}{n} e \right\| = 0.$$

A point $(\bar{x}, \bar{y})$ is in the neighbourhood of the central path if

$$(\bar{x}, \bar{y}) \in T(\theta) = \left\{ (x, y) \in \mathcal{F}^0 : \left\| XYe - \frac{x^T y}{n} e \right\| \leq \left(\frac{x^T y}{n} \right) \theta \right\},$$

where $\theta > 0$.

Algorithm

Input

$\epsilon > 0$, an accuracy parameter,

$\beta \in (0,1)$ and $0 < \theta \leq 0.1$ are given parameters;

$(x^0, y^0) \in T(\theta)$;

$k = 0$;

Begin

while $x^{kT} y^k \geq \epsilon$

compute $\mu = \frac{x^{kT} y^k}{n}$;

find $\left(\Delta x^k, \Delta y^k\right)$ solution of the following system

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

set $\left(x^{k+1}, y^{k+1}\right) = \left(x^k, y^k\right) + \left(\Delta x^k, \Delta y^k\right)$;

set $k = k + 1$;

end while

end.

Convergence

If the current point $\left(x^k, y^k\right)$ is close to the central path and a suitable value of the parameter β is chosen, then the new iterate $\left(x^{k+1}, y^{k+1}\right)$ remains close to the central path.

Concept of neighborhood

A path-following algorithm explicitly restricts the iterates to a neighborhood of the central path and follows this central path to a solution of the problem. The neighborhood excludes points that are too close to the boundary of the nonnegative orthant. Therefore, search directions calculated from any point in the neighborhood make at least minimal progress toward the solution set. The two most interesting neighborhoods of the central path are the so-called 2-norm neighborhood $\mathcal{N}_2(\theta)$ defined by

$$\mathcal{N}_2(\theta) = \left\{(x,y) \in \mathcal{F}^0 : \|XYe - \mu e\|_2 \leq \theta \mu\right\},$$

for some $\theta \in]0,1[$, and the ∞-norm neighborhood $\mathcal{N}_\infty^-(\gamma)$ defined by

$$\mathcal{N}_\infty^-(\gamma) = \left\{(x,y) \in \mathcal{F}^0 : x_i y_i \geq \gamma \mu \text{ for } i = 1,2,...,n\right\}$$

for some $\gamma \in]0,1[$. If a point lies in $\mathcal{N}_{\infty}^{-}(\gamma)$, each pairwise product $x_i y_i$ must be at least some small multiple γ of their average value μ. This requirement is actually quite modest, and we can make $\mathcal{N}_{\infty}^{-}(\gamma)$ encompass most of the feasible region $\mathcal{F}^0$ by choosing γ close to zero. The $\mathcal{N}_2(\theta)$ neighborhood is more restrictive, since certain points in $\mathcal{F}^0$ do not belong to $\mathcal{N}_2(\theta)$ no matter how close θ is chosen to its upper bound of 1.

Infeasible central path-following methods

One may distinguish between feasible IPMs and infeasible IPMs (IIPMs). Feasible IPMs start with a strictly feasible interior-point and maintain feasibility during the solution process. Feasible IPMs require that the starting points satisfy exactly the equality constraints and are strictly positive, i.e., they lie in the interior of a region defined by inequality constraints. All subsequent points generated by the feasible IPMs will have the same properties. IIPMs start with an arbitrary positive point and feasibility is reached as optimality is approached. The choice of the starting point in IIPMs is crucial for the performance.

Reconsider the LCP (1.3), replacing the equation $xy = 0$ by $xy = \mu e$, where $\mu > 0$, we get

$$F_{\mu}(x,y) = \begin{bmatrix} y - Mx - q \\ xy - \mu e \end{bmatrix} = 0.$$

To define the search direction we apply the Newton's method we get

$$\nabla F_{\mu}(x,y) \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = -F_{\mu}(x,y),$$

which is equivalent to

$$\begin{cases} \Delta y - M\Delta x = r_f \\ x\Delta y + y\Delta x = r_c, \end{cases} \tag{1.8}$$

where $r_f = Mx + q - y$ and $r_c = \mu e - xy$.

We start by choosing arbitrary $(x^0, y^0) > 0$, such that $x^0 y^0 = \mu^0 e$ for some positive number μ^0, we update this vector by computing a search direction and a suitable step-size $\alpha > 0$ in each iteration. We stop the algorithm when the

$$\max\left(\|y - Mx - q\|, \|xy\|\right)$$

is less than the accuracy parameter ϵ.

Algorithm

an accuracy parameter $\epsilon > 0$;

given $x^0,\ y^0 > 0$ and $\mu^0 > 0$;

set $k = 0$;

While $\max(\left\|y^k - Mx^k - q\right\|, \left\|x^k y^k\right\|) > \epsilon$ **do**

solve the system (1.8) to obtain $(\Delta x^k, \Delta y^k)$;

determine a step size $\alpha > 0\ : x^k + \alpha\Delta x^k > 0$ and $y^k + \alpha\Delta y^k > 0$;

update $x^{k+1} := x^k + \alpha\Delta x^k,\ y^{k+1} := y^k + \alpha\Delta y^k$;

set $k = k + 1$;

end.

Convergence

Theorem 1.3.5. *([56]) Let (x^k, y^k) be a sequence generated by the precedent algorithm, then for all $\epsilon > 0$ there exists k^*: $\max(\left\|y^k - Mx^k - q\right\|, \left\|x^k y^k\right\|) \leq \epsilon$, for all $k > k^*$.*

Interior-point method based on kernel functions

Most of interior-point methods for linear optimization are based on the logarithmic barrier function. In 2001, Peng et al. [47] proposed new variants of IPMs based on a new nonlogarithmic kernel functions. Such a function is strongly convex and smooth coercive on its domain. They obtained the best known complexity results for large and small-update methods.

Consider the LCP (1.3), we replace the equation $xy = 0$ by the parameterized equation $xy = \mu e$ with $\mu > 0$. Thus we consider the system

$$\begin{cases} y = Mx + q, \\ xy = \mu e, \\ x > 0, y > 0, \end{cases} \tag{1.9}$$

where M is positive semidefinite matrix. We suppose that $\mathcal{F}^0 \neq \emptyset$. Since M is a positive semidefinite matrix and LCP is strictly feasible, then (1.9) has a unique solution for any $\mu > 0$.

1/ **The classical approach**:

The classical way to define the search direction at the given point $x > 0, y > 0$ is to apply Newton's method to the nonlinear system (1.9). This yields the following system of linear equations for the

search directions Δx and Δy.

$$\begin{cases} M\Delta x = \Delta y \\ y\Delta x + x\Delta y = \mu e - xy. \end{cases}$$

By taking a step along the search direction with the step size defined by some line search rule, one constructs a new pair (x, y) that is closer to the μ-center $(x(\mu), y(\mu))$. Then, we reduce μ to $\mu^+ = (1-\theta)\,\mu$ for some fixed $\theta \in (0,1)$ and setting $\mu = \mu^+$, we apply Newton's method again, targeting at the new μ-center. Then μ is reduced again, and so on. This procedure is repeated until μ is small enough, say until $n\mu < \epsilon$.

Let (x, y) a strict feasible point and $\mu > 0$. Define

$$v = \sqrt{\frac{xy}{\mu}}.$$

Note that the pair (x, y) coincides with the $\mu-$center $(x(\mu), y(\mu))$ if and only if $v = e$.

Hence, a point on the central path can be characterized by the property $v_i = 1, \forall i$. In interior-point methods the iterates usually are not on the central path, but in some neighbourhood of it. A natural way to measure the deviation of the i^{th} coordinate v_i from one is to use the value at v_i of the smooth strictly convex function $\psi : \mathbb{R}_{++} \to \mathbb{R}_+$ that is nonnegative, assumes its minimal value zero at 1 and that goes to infinity when the argument goes to zero or infinity.

Let us recall the definition of the primal-dual logarithmic barrier function, which is given by

$$\phi_{lb}(xy, \mu) = \frac{x^T y}{\mu} - \sum_{i=1}^{n} \log \frac{x_i y_i}{\mu} - n,$$

this can be rewritten in terms of vector v as follows

$$\phi_{lb}(xy, \mu) = \sum_{i=1}^{n} (v_i^2 - 1 - log v_i^2),$$

hence, if the univariate function $\psi : \mathbb{R}_{++} \to \mathbb{R}_+$ is defined by

$$\psi_{lb}(t) = \frac{t^2 - 1}{2} - \log t.$$

We have $\phi_{lb}(xy, \mu) = 2\Psi_{lb}(v)$ where $\Psi_{lb}(v) = \sum_{i=1}^{n} \psi_{lb}(v_i)$.

A proximity function $\Psi(v)$, based on a function ψ, in the scaled space can be defined as follows

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i). \tag{1.10}$$

$\psi_{lb}(t)$ is nonnegative, strictly convex, minimal at $t = 1$ and $\psi_{lb}(1) = 0$. Consequently $\Psi_{lb}(v)$ is nonnegative, strictly convex, minimal at $v = e$ and $\Psi_{lb}(e) = 0$.

By introducing the notations

$$d = \sqrt{\frac{x}{y}},\ d_x = \frac{v\Delta x}{x},\ d_y = \frac{v\Delta y}{y}, \tag{1.11}$$

we can rewrite the previous system as the following scaled Newton-system:

$$\begin{cases} -\bar{M}d_x + d_y = 0 \\ d_x + d_y = v^{-1} - v, \end{cases}$$

where $\bar{M} = DMD$ and $D = diag\,(d)$. The right hand side in this equation is nothing else than the negative gradient of the scaled logarithmic barrier function $\Psi_{lb}(v)$. The second equation in this system is called the scaled centering equation which we replaced by

$$d_x + d_y = -\nabla\Psi\,(v)\,,$$

where $\Psi(v) = \Psi_{lb}(v)$. Therefore, we can get the following modified Newton-system:

$$\begin{cases} -\bar{M}d_x + d_y = 0 \\ d_x + d_y = -\nabla\Psi\,(v)\,, \end{cases} \tag{1.12}$$

Since M is a positive semidefinite matrix and LCP is strictly feasible, the search direction $(\Delta x, \Delta y)$ is uniquely defined. Using the proximity function $\Psi\,(v)$, in order to find the search direction and to measure the proximity between the current iterates and the μ-center, the generic primal-dual algorithm for LCPs can be described as follows.

Algorithm

Input

a proximity function $\Psi(v)$;

a threshold parameter $\tau > 0$ and an accuracy parameter $\epsilon > 0$;

a fixed barrier update parameter θ, $0 < \theta < 1$;

begin

$x = x^0; y = y^0; \mu = \mu^0;$

while $n\mu \geq \epsilon$ **do**

$\mu = (1 - \theta)\,\mu$;

$v = \sqrt{\frac{xy}{\mu}}$

while $\Psi(v) \geq \tau$ **do**

calculate $(\Delta x, \Delta y)$ from (1.12) and (1.11);

calculate a suitable step size α;

set $x = x + \alpha\Delta x$ and $y = y + \alpha\Delta y$;

end.

The inner "while loop" is called inner iteration and the outer "while loop" is called outer iteration. Each outer iteration consists of an update of parameter μ and a sequence of (one or more) inner iterations. The total number of inner iterations is referred to as iteration complexity of the algorithm. Usually this number is described as a function of the dimension n and ϵ.

2/ **The new approach**

In principle any nonnegative, strictly convex function satisfying

$$\psi'(1) = \psi(1) = 0, \psi'' > 0,\ \forall t > 0, \lim_{t\to 0} \psi(t) = \lim_{t\to\infty} = \infty,$$

can be used instead of the primal-dual logarithmic barrier function, then (1.10) can be used to define a new proximity function $\Psi(v)$ and the negative gradient of $\Psi(v)$ can be used to define new search direction. A key task in designing interior-point methods is to find proper functions that are suitable to design and analyze polynomial interior-point methods.

Predictor-corrector method for LCP

The prediction-correction name comes from the domain of algorithms for solving differential equations, there is a prediction step and a correction step. The principle of the prediction-correction method is to try to achieve two objectives: to remain central and to get closer to the optimum by

means of two different types of iterations.

Going back to the problem (1.3), relaxing the complementarity condition we get

$$\begin{cases} XYe = \sigma\mu e, \\ y = Mx + q, \\ (x, y) > 0. \end{cases} \tag{1.13}$$

where $\mu > 0$ and $\sigma \in [0, 1]$ is the reducing factor of duality measure μ.

Let

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

Then solving (1.13) is equivalent to solve the nonlinear system

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

Applying Newton's method we get

$$\begin{cases} Y\Delta x + X\Delta y = \sigma\mu e - XYe \\ -M\Delta x + \Delta y = 0. \end{cases} \tag{1.14}$$

Predictor-corrector method uses the two directions an affine direction (prediction) to reduce μ and a centring direction (correction) to stay close to central path.

Affine direction

Let $\sigma = 0$, then we solve the system

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

We get the so-called predictor search directions noted $(\Delta x^a, \Delta y^a)$.

Centering direction

Let $\sigma = 1$, then we solve the system

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

The obtained directions called corrector search directions noted $(\Delta x^c, \Delta y^c)$.

Mehrotra's predictor-corrector algorithm

In 1989, Mehrotra proposed the first predictor-corrector algorithm to solve a linear program that is very efficient numerically. His idea was to determine the centering parameter in an adaptive way and no longer according to a predetermined value as in the central path-following algorithms. Several variants of this algorithm were introduced for linear programming and were later extended to LCP by several researchers.

In the following, we describe a feasible version of the original Mehrotra's algorithm. Note that the original algorithm of Mehrotra is infeasible.

In the prediction step the algorithm calculates the affine direction $(\Delta x^a, \Delta y^a)$ by setting $\sigma = 0$ in (1.14) that is

$$\begin{cases} M\Delta x^a = \Delta y^a \\ x\Delta y^a + y\Delta x^a = -xy. \end{cases} \tag{1.15}$$

Then we find the longest step length $\alpha_a \in [0, 1]$ which guarantees

$$(x + \alpha_a \Delta x^a, y + \alpha_a \Delta y^a) \geq 0. \tag{1.16}$$

We define μ^a to be the value of μ that would be obtained by a full step to the boundary, that is

$$\mu^a = \frac{(x + \alpha_a \Delta x^a)^T (y + \alpha_a \Delta y^a)}{n},$$

and set the centering parameter to be

$$\sigma = \left(\frac{\mu^a}{\mu}\right)^3.$$

We can see that when good progress is made along the predictor direction we have $\mu^a << \mu$, so the σ obtained from this formula is small and conversely.

Using the outcomes of (1.15) and (1.16) we obtain the corrector direction $(\Delta x, \Delta y)$ by solving the system

$$\begin{cases} M\Delta x = \Delta y \\ x\Delta y + y\Delta x = \sigma\mu e - \Delta x^a \Delta y^a e - xy, \end{cases} \tag{1.17}$$

the new iterate is

$$(x, y) = (x, y) + \alpha\,(\Delta x, \Delta y)\,.$$

where

$$\alpha_x = \min(1, \min_{i:\Delta x_i<0} \frac{-x_i}{\Delta x_i}), \alpha_y = \min(1, \min_{i:\Delta y_i<0} \frac{-y_i}{\Delta y_i}), \alpha = \min(1, \min(\beta\alpha_x, \beta\alpha_y)), \tag{1.18}$$

and $\beta \in [0.9, 1[$.

Algorithm

Input

an accuracy parameter $\epsilon > 0$;

a proximity parameter $\gamma \in (0, 1)$;

given $(x^0, y^0) \in \mathcal{N}_\infty^-(\gamma)$;

for $k = 0, 1, ...$

set $(x, y) = (x^k, y^k)$ and solve (1.15) to get $(\Delta x^a, \Delta y^a)$;

calculate α^a and μ^a;

set $\sigma = \left(\frac{\mu^a}{\mu}\right)^3$;

solve (1.17) for $(\Delta x, \Delta y)$;

calculate α from (1.18);

set $x^{k+1} = x^k + \alpha_k \Delta x^k$, $y^{k+1} = y^k + \alpha_k \Delta y^k$;

end.

1.4 Some application examples of LCPs

Complementarity problems arise naturally in the study of many phenomena in economics and engineering. A comprehensive and excellent treatment of applications of complementarity problems is provided in [25]. Additionally, a large collection of problems from a variety of application areas can be found in the MCPLIB library of test problems [15]. Applications of complementarity from the field of economics include general Walrasian equilibrium, spatial price equilibria, invariant capital stock, and game-theoretic models. In engineering, complementarity problems arise in contact mechanics, structural mechanics, obstacle and free boundary problems, elastohydrodynamic lubrication, and trafic equilibrium.

As a rule of thumb, the complementarity framework should be considered whenever the system being studied involves complementary pairs of variables (that is, where one or the other member of each pair must be at its bound).

Linear complementarity problems also have many applications in mathematical programming and equilibrium problems. These applications are too numerous to be listed all. For example, optimization problems (linear programming, convex quadratic programming,...), bimatrix and equilibrium points, variational inequalities, fixed point problems and set of piecewise-linear equations...,for more details, basic theory, algorithms, and applications, the reader can refer to [20].

We cite here some LCPs applications:

1.4.1 Quadratic programming formulations

We consider the quadratic program

$$\begin{cases} minimize \quad f(x) = q^T x + x^T M x, \\ x \geq 0, \end{cases} \tag{1.19}$$

If M is positive semidefinite, the program (1.19) is completely equivalent to the LCP, where M is symmetric.

A significant number of applications in engineering and the physical sciences lead to a convex quadratic programming model of the special type (1.19) which is equivalent to the LCP.

These applications include the contact problem, the porous flow problem, the obstacle problem, the journal bearing problem, the elastic-plastic torsion problem as well as many other free-boundary problems. A common feature of these problems is that they are all posed in an infinite-dimensional function space setting. The quadratic program (1.19) to which they give rise is obtained from their finite-dimensional discretization. Consequently, the size of the resulting program tends to be very large. The LCP plays an important role in the numerical solution of these applied problems.

1.4.2 Fixed-point formulations

A fixed-point of a mapping $h : \mathbb{R}^n \to \mathbb{R}^n$ is a vector z such that $z = h(z)$.

Obviously, finding a fixed-point of the mapping h is equivalent to finding a zero of the mapping $g(z) = z - h(z)$. Conversely, finding a zero of the mapping $g : \mathbb{R}^n \to \mathbb{R}^n$ is easily translated into finding a fixed point of the mapping $h(z) = z - g(z)$. We describe several ways of transforming a linear complementarity problem into either a fixed-point or zero-finding problem.

The simplest zero-finding formulation of the LCP with given M and q is gotten by defining

$$g(z) = min(z, q + Mz). \tag{1.20}$$

Obviously, a vector z is a solution of the LCP if and only if $g(z) = 0$. The corresponding fixed-point formulation is defined by the mapping

$$h(z) = z - g(z) = max(0, -q + (I - M)z), \tag{1.21}$$

Note that $h(z)$ can be interpreted as the projection of $-q + (I - M)z$ onto the nonnegative orthant. There are several variations of the formulations (1.20) and (1.21). We mention one which is obtained by scaling the vectors z and $w = q + Mz$.

Let D and E be two $n \times n$ diagonal matrices with positive diagonal entries. Define

$$g(z) = min(Dz, E(q + Mz)). \tag{1.22}$$

Again, a vector z is a solution of the LCP if and only if $g(z) = 0$. In this case the associated fixed-point mapping

$$h(z) = z - g(z)$$

can no longer be interpreted as a projection.

1.4.3 Piecewise linear equations

Given $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$, find $x, y \in \mathbb{R}^n$ satisfying

$$y = Mx + q, x^T y = 0, \quad x \geq 0, y \geq 0, \tag{1.23}$$

for each $x \in \mathbb{R}^n$ define

$$h_i(x) = min(x_i, (q + Mx)_i),$$

and let

$$h(x) = (h_i(x)).$$

Then $h(x) : \mathbb{R}^n \to \mathbb{R}^n$ is a piecewise linear concave function, and clearly, solving the LCP (1.23) is equivalent to solve the system of piecewise linear equations

$$h(x) = 0.$$

Conversely, under a mild nonsingularity assumption, any piecewise linear system of equations can be reformulated as a linear complementarity problem.

1.4.4 Obstacle and free boundary problems

In this simple obstacle problem, we try to compute a trajectory joining the boundary of a domain with an obstacle g, and a minimal curvature f. This can be formulated using the following equation and

inequalities: find u such that

$$\left(\ddot{u}\left(x\right)-f\left(x\right)\right)^{T}\left(u\left(x\right)-g\left(x\right)\right)=0,\ddot{u}\left(x\right)-f\left(x\right)\geq 0,u\left(x\right)-g\left(x\right)\geq 0.$$

We approximate the second order derivative with a second-order centred finite difference to get a discrete version on an equispaced grid $x_i = ih, i = 1, ..., n$.

$$\left(Du-f\right)^{T}\left(u-g\right)=0,Du-f\geq 0,u-g\geq 0,$$

where

$$D=\begin{bmatrix} \frac{2}{h^2} & \frac{-1}{h^2} & & \\ \frac{-1}{h^2} & \ddots & \ddots & \\ & \ddots & \ddots & \frac{-1}{h^2} \\ & & \frac{-1}{h^2} & \frac{2}{h^2} \end{bmatrix}, g_i = g\left(x_i\right) \text{ and } f_i = f\left(x_i\right).$$

This can be written as a linear complementarity problem by setting

$$z=u-g,M=D \text{ and } q=Dg-f,$$

that is

$$\left(Mz+q\right)^{T}z=0,Mz+q\geq 0,z\geq 0.$$

Chapter 2

Horizontal linear complementarity problem(HLCP)

2.1 Problem statement

A slightly more general problem is the horizontal linear complementarity problem (HLCP), which seeks x and y satisfying

$$Ny - Mx = q,\ x \geq 0,\ y \geq 0,\ x^T y = 0, \tag{2.1}$$

where $N, M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ are given. We denote

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : Ny - Mx = q, x \geq 0, y \geq 0\}$$

the set of feasible solutions of (2.1), and

$$\mathcal{F}^0 = \{(x, y) \in \mathcal{F} : x > 0, y > 0\}$$

the set of its strict feasible solutions. If N is invertible, then the HLCP is reduced to an LCP.

Classes of pairs

- (M, N) is called a **monotone pair** if $-Mx + Ny = 0$ implies $x^T y \geq 0$.
- (M, N) is called a P_0**-pair** (respectively a P**-pair**) if $-Mx + Ny = 0$ and $(x, y) \neq 0$ implies that there exists an index i such that $x_i \neq 0$ or $y_i \neq 0$ and $x_i y_i \geq 0$ (respectively, such that $x_i y_i > 0$). Note that P-pairs are thus also P_0-pairs.

- (M, N) is called a $P_*(\kappa)$**-pair** if

$$-Mx + Ny = 0 \Rightarrow (1 + 4\kappa) \sum_{i \in I^+(x)} x_i y_i + \sum_{i \in I^-(x)} x_i y_i \geq 0,$$

where

$$\kappa \geq 0, I^+(x) = \{i \in I : x_i y_i \geq 0\}, I^-(x) = \{i \in I : x_i y_i < 0\}, \text{ and } I = \{1, 2, ..., n\}.$$

If $\kappa = 0$, we get the class of monotone pairs.

The HLCP defined by a monotone pair is solvable whenever it is feasible.

We can see that LCP $\subset$ HLCP.

LCP is not just a special case of HLCP but is actually equivalent to this formulation, that is: HLCP=LCP.

The messing links that we need to complete the proof of this result are given in the following theorem:

Theorem 2.1.1. *(Theorem 8.2 in [58]) ($HLCP \subset LCP$) Any horizontal linear complementarity problem can be reformulated as an LCP. Primal-dual algorithms applied to the two formulations generate sequences of iterates $\{(x^k, y^k)\}$ that are identical to within a permutations of the variables.*

The proof of Theorem 2.1.1 can be found in [13].

2.2 Solving absolute value equations using complementarity and interior-point methods

In this part, an infeasible path-following interior-point algorithm is proposed for solving the absolute value equation across the HLCP [2]. This equation is manifested in differential and partial derivative equations.

Given two matrices $A, B \in \mathbb{R}^{n \times n}, B \neq 0$, and a vector $b \in \mathbb{R}^n$, the absolute value equations (AVE) is given by

$$Ax - B|x| = b. \tag{2.2}$$

Here, $|x|$ denotes the componentwise absolute value of vector $x \in \mathbb{R}^n$.

Reformulation of the AVE as an HLCP

Definition 2.2.1. *For $x \in \mathbb{R}^n$ the vectors x^+ and x^- are defined such that*

$$x_i^+ = \max_i(0, x_i) \text{ and } x_i^- = \max_i(0, -x_i).$$

Then

$$x^+ \geq 0,\ x^- \geq 0,\ x = x^+ - x^-,\ |x| = x^+ + x^-,\ (x^+)^T x^- = 0. \tag{2.3}$$

According to the decompositions of x and $|x|$ (2.3), the AVE in (2.2) can be reduced as the following equivalent HLCP:

$$\begin{cases} Nx^+ - Mx^- = q \\ x^+ \geq 0,\ \ x^- \geq 0 \\ x^{+T} x^- = 0 \end{cases} \tag{2.4}$$

where

$$N = (A - B),\ M = (A + B) \text{ and } q = b.$$

Theorem 2.2.1 (Theorem 3.3.7 in [20])**.** *A matrix $M \in \mathbb{R}^{n\times n}$ is a P-matrix if and only if the corresponding LCP has a unique solution for every $q \in \mathbb{R}^n$.*

In what follows we assume that the AVE (2.2) satisfy the condition

- **Assumption*.** The pair of the matrices $[A, B]$, satisfies

$$\sigma_{\min}(A) > \sigma_{\max}(B).$$

From the theory of the HLCP, and under **Assumption***, we show, based on the following theorem, that the AVE (2.2) is uniquely solvable for every $b \in \mathbb{R}^n$.

Theorem 2.2.2. *Under **Assumption***, the AVE (2.2) is uniquely solvable for every $b \in \mathbb{R}^n$.*

Proof. We prove under **Assumption***, that HLCP in (2.4), is reduced to a standard P-LCP with $M = (A - B)^{-1}(A + B)$ and $q = (A - B)^{-1}b$. Then we prove only that M is positive definite. We show first that the matrix $(A - B)$ is invertible. If not, for some nonzero $x \in \mathbb{R}^n$, we have that $(A - B)x = 0$, which derives a contradiction. This implies that $Ax = Bx$. Hence,

$$\begin{aligned} \sigma_{\min}(A) &= \min_{\|z\|=1} \langle A^T Az, z\rangle \leq \langle A^T Ax, x\rangle = \langle B^T Bx, x\rangle \\ &\leq \max_{\|z\|=1} \langle B^T Bz, z\rangle = \sigma_{\max}(B) \end{aligned}$$

which contradicts our condition. Hence $(A-B)^{-1}$ is invertible. Now, we prove that M is positive definite matrix. To this end, we have, for all nonzero $x \in \mathbb{R}^n$,

$$\langle (A-B)^{-1}(A+B)x, x\rangle = \langle (A+B)x, (A^T-B^T)^{-1}x\rangle.$$

Letting $(A^T-B^T)^{-1}x = z$, one has

$$\begin{aligned}\langle (A-B)^{-1}(A+B)x, x\rangle &= \langle (A+B)(A^T-B^T)z, z\rangle \\ &= \langle (AA^T-BB^T-AB^T+BA^T)z, z\rangle \\ &= \langle (AA^T-BB^T)z, z\rangle + \langle (BA^T-AB^T)z, z\rangle.\end{aligned}$$

Note that $\langle (BA^T-AB^T)z, z\rangle = 0$. $(A-B)^{-1}(A+B)$ is positive definite if and only if (AA^T-BB^T) is positive definite. Indeed, by Weyl's inequalities [Theorem 4.3.1 in [30]], we find that

$$\lambda_{\min}(AA^T-BB^T) \geq \lambda_{\min}(AA^T) + \lambda_{\min}(-BB^T) = \lambda_{\min}(AA^T) - \lambda_{\max}(BB^T).$$

From **Assumption***, $\lambda_{\min}(AA^T) - \lambda_{\max}(BB^T) > 0$, $(AA^T - BB^T)$ is positive definite. Hence $(A-B)^{-1}(A+B)$ is positive definite. Consequently, M is a P-matrix. Thus from Theorem 2.2.1, the standard LCP has a unique solution for any $q \in \mathbb{R}^n$, so is the AVE (2.2) for any $b \in \mathbb{R}^n$. This achieves the proof. □

Corollary 2.2.1. *The vector $x^* = x_*^+ - x_*^-$ is the unique solution of AVE (2.2) if and only if the pair of vectors $(x_*^+,\ x_*^-)$ is the unique solution of HLCP (2.4).*

Proof. Note that the AVE (2.2) is uniquely solvable for every b. Let x^* be its unique solution, i.e., x^* satisfies

$$Ax^* - B\,|x^*| = b.$$

From (2.3), $x^* = x_*^+ - x_*^-$ and $|x^*| = x_*^+ + x_*^-$, we arrive at

$$A(x_+^* - x_-^*) - B(x_+^* + x_-^*) = b,$$

which is equivalent to

$$(A-B)x_+^* - (A+B)x_-^* = b.$$

Let $A-B=N, A+B=M$ and $q=b$. Hence $Nx_+^* - Mx_-^* = q$. From (2.3), we have $x_+^*, x_-^* \geq 0$ and $\langle x_+^*, x_-^*\rangle = 0$. Consequently, the pair (x_+^*, x_-^*) is a solution of HLCP (2.4). Note that the HLCP

is reduced to an P-LCP. From Theorem 2.2.1, we see that this solution is unique. Conversely, with the same manner, we can prove the inverse implication. This completes the proof. □

Solving AVE by interior-point methods

In this part, we need to put the AVE (2.2) into the framework of interior-point methods. It's known that solving the HLCP (2.4) is equivalent to solve the following nonlinear system of equations

$$\begin{cases} Nx^+ - Mx^- = q, x^+ \geq 0, \ x^- \geq 0 \\ X^+x^- = 0, \end{cases} \tag{2.5}$$

where X^+:= diag(x^+). Next, the feasible set and the strict feasible set of HLCP are, respectively, denoted by

$$\mathcal{F} = \left\{(x^+, x^-) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : \ Nx^+ = Mx^- + q\right\},$$

and

$$\mathcal{F}^0 = \left\{(x^+, x^-) \in \mathcal{F}, \ x^+ > 0, \ x^- > 0\right\}.$$

Besides the strict monotony of HLCP (2.4), we assume in the sequel that $\mathcal{F}^0 \neq \emptyset$, this assumption is known as the interior-point condition (IPC). These two conditions play together an important role for the analysis of the convergence of infeasible interior-point algorithms for solving HLCP (see, e.g. [61, 58]). The path-following interior-point methods [58], are based on replacing the second equation in (2.5) by the parameterized equation $X^+x^- = \mu e$ where $\mu > 0$. Thus we consider the following perturbed system

$$F_\mu(x^+, x^-) = \begin{bmatrix} Nx^+ - Mx^- - b \\ X^+x^- - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.6}$$

Under the condition of IPC, (2.6) has a unique solution denoted by $((x^+(\mu), x^-(\mu))$ for all $\mu > 0$. The set

$$\mathcal{C} = \left\{(x^+(\mu), x^-(\mu)) : \mu > 0\right\}$$

is called the central-path of HLCP. If $\mu \to 0$, then the limit of the central-path exists and since the limit point satisfies the complementarity condition, the limit yields a solution of HLCP [61]. Now, we are in a position to define search directions $(\Delta x^+, \Delta x^-)$ that move in the direction of the central-path $\mathcal{C}$. Applying Newton's method for (2.6), we get

$$\nabla F_\mu(x^+, x^-) \begin{bmatrix} \Delta x^+ \\ \Delta x^- \end{bmatrix} = -F_\mu(x^+, x^-)$$

which is equivalent to the following linear system

$$\begin{cases} N\Delta x^+ - M\Delta x^- = r_f \\ X^-\Delta x^+ + X^+\Delta x^- = r_c \end{cases} \tag{2.7}$$

where X^-=Diag(x^-), $r_f = Mx^- + b - Nx^+$and $r_c = \mu e - X^+x^-$. Note that if x^+ and x^- are in $\mathcal{F}^0$ i.e., are strictly feasible then $r_f = 0$. The unique solution of (2.7), is guaranteed by assumptions $\mathcal{F}^0 \neq \emptyset$ and the strict monotony of HLCP since the block matrix

$$\begin{bmatrix} N & -M \\ X^- & X^+ \end{bmatrix}$$

is invertible (see Proposition 3.1 in [58]).

An infeasible path-following algorithm for AVE. We present an infeasible path-following interior-point algorithm for computing the unique solution of HLCP in (2.4) that uses the primal-dual interior-point framework proposed by many authors. In each iteration of the algorithm we start with guesses $x^+, x^- > 0$, not necessarily feasible. We would like to update these vectors until we are within our desired tolerance of satisfying equations in (2.5). We stop our algorithm when the

$$\max\left(\left\|Nx^+ - Mx^- - b\right\|, \left\|X^+x^-\right\|\right)$$

is small enough.

Algorithm

Input:

An accuracy parameter $\epsilon > 0$;

initial guesses $x^+,\ x^- > 0$ and $\mu > 0$;

two matrices A and B, a vector b with $N = A - B$ and $M = A + B$;

While $\max(\|Nx^+ - Mx^- - b\|, \|X^+x^-\|) > \epsilon$ **do**

begin

- Solve the system (2.7) to obtain $(\Delta x^+, \Delta x^-)$;

- Determine a step size $\alpha > 0$ s.t. $x^+ + \alpha\Delta x^+ > 0$ and $x^- + \alpha\Delta x^- > 0$;

- Update $x^+ := x^+ + \alpha\Delta x^+,\ x^- := x^- + \alpha\Delta x^-$;

end

The direction is determined by solving the linear system in (2.7). We compute the damped Newton step-size $\alpha > 0$ so that $x^+ + \alpha\Delta x^+ > 0$ and $x^- + \alpha\Delta x^- > 0$. We need to determine a maximum possible step-size $\alpha_{\max}$ so that if $0 < \alpha < \alpha_{\max}$ then $x^+ + \alpha\Delta x^+ > 0$ and $x^- + \alpha\Delta x^- > 0$. Let α_{x^+} and α_{x^-} be the maximum possible step-sizes in the directions Δx^+ and Δx^-, respectively. Then our maximum step-size is $\alpha_{max} = min\,(\alpha_{x^+}, \alpha_{x^-})$, where

$$\alpha_{x^+} = \begin{cases} \min\limits_i \left(-\frac{x_i^+}{\Delta x_i^+}\right) & \text{if } \left(\Delta x_i^+\right) < 0, \forall i = 1, ..., n \\ 1 & \text{otherwise} \end{cases}$$

and

$$\alpha_{x^-} = \begin{cases} \min\limits_i \left(-\frac{x_i^-}{\Delta x_i^-}\right) & \text{if } \left(\Delta x_i^-\right) < 0, \forall i = 1, ..., n \\ 1 & \text{otherwise} \end{cases}$$

Then, we compute our step-size as $\alpha = \min(\rho\alpha_{\max}, 1)$, for some $\rho \in (0, 1]$, which guaranties that our next iterates will be positive. For specifying the parameter $\mu > 0$, it is easily seen from the second equation in (2.6) that $\mu = \frac{x^{+T}x^-}{n}$ which is known as the duality gap of HLCP. The polynomial complexity of these algorithms is proved by many authors in the literature of infeasible interior-point methods (see, e.g., [61, 58]).

Numerical results

In this part, the pair (x_0^+, x_0^-) denotes the infeasible starting point for the algorithm and the pair (x_*^+, x_*^-) denotes the unique solution of the HLCP. The number of iterations and the solution for the AVE (2.2) are denoted by "**It**" and $x^* = x_*^+ - x_*^-$, respectively. The implementation of the algorithm is running on MATLAB 7.1 and our accuracy in all examples is set to $\epsilon = 10^{-6}$ and the parameter ρ is chosen such that $0.85 \leq \rho \leq 0.95$.

Problem 2.1. The matrices A and B and $b \in \mathbb{R}^n$ are given by

$$A = (a_{ij}) = \begin{cases} 5 & \text{if} \quad i = j \\ 1 & \text{if} \quad |i - j| = 1 \quad \forall i, j = 1, .., n \\ 0 & \text{otherwise} \end{cases}$$

$$B = (b_{ij}) = \begin{cases} 1 & \text{if} \quad i = j \\ 1 & \text{if} \quad |i - j| = 1 \quad \forall i, j = 1, .., n \\ 0 & \text{otherwise} \end{cases}$$

and

$$b = (-8,\ -10, ..., -10,\ -8)^T \in \mathbb{R}^n.$$

The pair of initial point for this problem is:

$$x_0^- = x_0^+ = (1,\ 2,\ ...,\ n)^T,$$

and the exact solution is $x^* = e$. The algorithm gives an approximate solution of the problem 2.1 as

$$x = (0.99995,\ 0.99995,\ ...,\ 0.99995)^T.$$

The numerical results for different sizes of n are summarized in table (2.1).

n	**It**	**Time(s)**
3	22	0.0510
10	19	0.0592
100	15	0.8542
500	12	39.0083
1000	11	248.945

Table 2.1: Numerical results of problem 2.1.

Problem 2.2. The data (A, B, b) of the AVE is given by

$$A = (a_{ij}) = \begin{cases} 10 & \text{if} \quad i = j \\ -1 & \text{if} \quad |i-j| = 1 \quad \forall i, j = 1, .., n \\ 0 & \text{otherwise} \end{cases}$$

$$B = (b_{ij}) = \begin{cases} 5 & \text{if} \quad i = j \\ -1 & \text{if} \quad |i-j| = 1 \quad \forall i, j = 1, .., n \\ 0 & \text{otherwise} \end{cases}$$

and $b = (A - I)\, e$. The exact solution is

$$x^* = (1.6, 1.4, ..., 1.4, 1.6)^T.$$

The numerical results are summarized in table (2.2). The algorithm gives an approximate solution of the problem 2.2 as

$$x = (1.59999,\ 1.39999,\ ...,\ 1.59999)^T.$$

n	**It**	**Time(s)**
3	21	0.0506
10	20	0.0640
100	18	0.956
520	16	62.240
1000	16	385.66

Table 2.2: Numerical results of problem 2.2.

Problem 2.3 The AVE is given by

$$A = \begin{pmatrix} 1.5 & 0.5 & 0 & 1.5 & -1.5 \\ 0.5 & 2.5 & 0 & 2 & 1 \\ 0 & 0 & 3.5 & -1 & 0.5 \\ -1.5 & -2 & 1 & 0.5 & 0 \\ 1.5 & -1 & -0.5 & 0 & 0.5 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.5 & 0.5 & 0 & 1.5 & -1.5 \\ 0.5 & 1.5 & 0 & 2 & 1 \\ 0 & 0 & 2.5 & -1 & 0.5 \\ -1.5 & -2 & 1 & -0.5 & 0 \\ 1.5 & -1 & -0.5 & 0 & -0.5 \end{pmatrix}$$

and

$$b = (1, -2, 4, 10, 2)^T .$$

The exact solution is $x^* = (1.5, -0.5, 4, 8, 1)^T$. So the number of iterations is 14 in 0.23 seconds.

Problem 2.4 The AVE is given by

$$A = \begin{pmatrix} 6 & 0.5 & 0.5 & \cdots & 0.5 & 0 \\ 0.5 & 6 & 0.5 & \cdots & 0.5 & 0 \\ 0.5 & 0.5 & 6 & \cdots & 0.5 & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & 0 \\ 0.5 & 0.5 & 0.5 & \cdots & 6 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 6 \end{pmatrix}, \; B = \begin{pmatrix} -1 & 0.5 & 0.5 & \cdots & 0.5 & 0 \\ 0.5 & -1 & 0.5 & \cdots & 0.5 & 0 \\ 0.5 & 0.5 & -1 & \cdots & 0.5 & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & 0 \\ 0.5 & 0.5 & 0.5 & \cdots & -1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix}$$

and

$$b = (21, 28, \cdots, 28, 21)^T$$

The exact solution is $x^* = (3, 4, \cdots, 4, 3)^T$. The algorithm gives an approximate solution of the problem 2.4 as $x = (3.000004,\ 4.000004,\ ...,\ 4.000004, 2.99999)^T$. The following table contains numerical results for this AVE

n	**It**	**Time (s)**
5	20	0.005
50	17	0.45
100	16	0.54
1000	12	220.86
1500	11	728.11

Table 2.3: Numerical results of problem 2.4.

In the next example, we compare our algorithm with an other feasible primal-dual IPMs used to solve the AVE (2.2).

Problem 2.5 [60]. The data (A, B, b) of the AVE is given by

$$\begin{aligned}
&rand('state', 0); R = rand(n, n);\\
&A = R' * R + n * eye(n); b = rand(n, 1);\\
&B = I;\\
&q = ((A + eye(n)) * (inv(A - eye(n))) - eye(n)) * b;
\end{aligned}$$

The numerical results are summarized in table (2.4), where our algorithm is denoted by **Algo1** and the algorithm used in [60] by **Algo2**. From [60], we use the same tolerance $\epsilon = 10^{-4}$.

Size	Algorithms	**It**	**Time(s)**
8	**Algo**1	13	0.1103
	Algo2	19	8.1350
32	**Algo**1	12	0.1445
	Algo2	34	9.1270
128	**Algo**1	10	0.3802
	Algo2	94	15.0780
512	**Algo**1	7	18.3298
	Algo2	147	21.6560
1024	**Algo**1	5	94.8895
	Algo2	171	37.9070

Table 2.4: Numerical results of problem 2.4.

From the numerical results stated in the above tables, we see that the algorithm offers a solution of the absolute value equations. In addition, the numerical results, obtained for problem 2.5, confirm that the algorithm performs well in comparison with those obtained via the feasible interior-point methods since the number of iterations and times are almost less. From above tables, we also mention that the advantage of interior-point methods for solving this problem is that for the higher dimension of the problem, the lower number of iterations is obtained.

Chapter 3

A primal-dual interior point method for $P_*(\kappa)$-HLCP based on a class of parametric kernel functions

In this chapter, we are interested in the central path-following methods via a kernel function. Currently, researchers are trying to improve the algorithmic complexity of interior-point methods, using new techniques, knowing that these approaches have the advantage of starting the algorithm with any starting point (x^0, y^0) strictly feasible. We consider the HLCP in the standard form as follows

$$-Mx + Ny = q, \quad xy = 0, \quad (x, y) \geq 0, \tag{3.1}$$

where $M, N \in \mathbb{R}^{n\times n}, x, y, q \in \mathbb{R}^n$ and xy denotes the componentwise product of vectors x and y, with

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : Ny - Mx = q, x \geq 0, y \geq 0\},$$

$$\mathcal{F}^0 = \{(x, y) \in \mathcal{F} : x > 0, y > 0\}.$$

The aim of this chapter is to propose a primal-dual IPM based on a new class of kernel functions for solving the HLCP. In all that follows, we have these assumptions:

- **Assumption 1** Without loss of generality we may assume that the HLCP satisfies the interior-point condition, i.e., there exists a pair of vectors (x^0, y^0) such that

$$-Mx^0 + Ny^0 = q, \left(x^0, y^0\right) > 0.$$

- **Assumption 2** The pair $\{M, N\}$ is a $P_*(\kappa)$-pair.

The basic idea of primal-dual IPM is to replace the second equation in (3.1) by the nonlinear equation $xy = \mu e$, where e is the all-one vector, and $\mu > 0$. Thus we have the following parameterized system:

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

Under **Assumption 1** and **Assumption 2 ,** the parameterized system (3.2) has a unique solution $(x(\mu), y(\mu))$ for each $\mu > 0$. $(x(\mu), y(\mu))$ is called $\mu-$center of (3.2), the set of $\mu-$centers $(\mu > 0)$ defines a homotopy path, which is called the central path of (3.2). If $\mu \to 0$, the limit of the central path exists. This limit satisfies the complementarity condition, and belongs to the solution set of (3.1), [37]. Using Newton's method to the system (3.2), we have

$$\begin{cases} -M\Delta x + N\Delta y = 0 \\ Y\Delta x + X\Delta y = \mu e - xy, \end{cases} \tag{3.3}$$

where $X = diag\,(x)$ and $Y = diag\,(y)$.

For the formulation and analysis of the generic interior-point method for the HLCP, the introduction of the following vector is critical :

$$v = \sqrt{\frac{xy}{\mu}}, \tag{3.4}$$

where $(x, y) \geq 0$ and $\mu > 0$. Note that the point (x, y) coincides with the μ-center $(x\,(\mu)\,, y\,(\mu))$ if and only if $v = e$. Next, the following scaled directions are introduced:

$$d_x = \frac{v\Delta x}{x}, d_y = \frac{v\Delta y}{y}. \tag{3.5}$$

Then we have the scaled Newton system as follows

$$\begin{cases} -\bar{M}d_x + \bar{N}d_y = 0 \\ d_x + d_y = v^{-1} - v, \end{cases} \tag{3.6}$$

where $\bar{M} = DMD$ and $\bar{N} = DND$ with $D = X^{\frac{1}{2}}Y^{\frac{1}{2}}, X = diag(x), Y = diag(y)$. Note that the right-hand side $v^{-1} - v$ in the second equation of (3.6) is equal to the negative gradient of the classical logarithmic barrier function $\Psi_c\,(v)$, i.e.,

$$d_x + d_y = -\nabla\Psi_c\,(v)\,,$$

where $\Psi_c(v) = \sum_{i=1}^{n} \psi_c(v_i)$ and $\psi_c(t) = \frac{t^2-1}{2} - \log t$ with $t > 0$ (v_i represents the i^{th} component of the vector v). We call $\psi_c(t)$ the kernel function of the classical logarithmic barrier function $\Psi_c(v)$.

Now, we consider the barrier function

$$\Psi(v) = \Psi(x, y; \mu) = \sum_{i=1}^{n} \psi(v_i),$$

where the univariate function $\psi(t) : D \to \mathbb{R}_+$ with $D \subseteq \mathbb{R}_{++}$ is called the kernel function of $\Psi(v)$ if $\psi(t)$ is twice differentiable and satisfies the following conditions

$$\psi(1) = \psi'(1) = 0, \quad \psi''(t) > 0, \quad \lim_{t\to 0} \psi(t) = \lim_{t\to\infty} \psi(t) = \infty.$$

Hence $\Psi(v)$ is strictly convex and minimal at $v = e$, i.e.,

$$\Psi(v) = 0 \Leftrightarrow \nabla\Psi(v) = 0 \Leftrightarrow v = e.$$

We can replace $\Psi_c(v)$ by the strictly convex barrier function $\Psi(v)$ such that the new scaled centering equation becomes

$$d_x + d_y = -\nabla\Psi(v). \tag{3.7}$$

Then the system (3.3) can be written as the following modified Newton system

$$\begin{cases} -M\Delta x + N\Delta y = 0 \\ x\Delta y + y\Delta x = -\mu v \nabla\Psi(v). \end{cases} \tag{3.8}$$

The algorithm works as follows. Suppose that the current iterate (x, y) is known and is in the τ-neighborhood of the corresponding μ-center, i.e., $\Psi(v) \leq \tau$. Next, the value of μ is reduced by the factor $1 - \theta$ with $0 < \theta < 1$, which changes the value of v according to (3.4) and defines a new μ-center $(x(\mu), y(\mu))$. This will cause the increase of the value of the barrier function above the threshold value of τ, i.e., $\Psi(v) \geq \tau$. Now we start the inner iteration by calculating new iterates where Δx and Δy are calculated from (3.8) and the step size α is chosen appropriately with the goal of reducing the value of barrier function $\Psi(v)$. If necessary, the procedure is repeated until we find the iterate that again belongs to the τ-neighborhood of the current μ-center, that is, until $\Psi(v) \leq \tau$. It is important to mention that during the inner iteration the value of μ is kept constant. At this point we start a new outer iteration by reducing the value of μ again. This process is repeated until μ is small enough, say until $n\mu < \epsilon$ for a certain (small) value of the accuracy parameter $\epsilon > 0$.

Algorithm: Generic Primal-Dual Algorithm for $P_*(\kappa)$-HLCPs

Input :

a proximity function $\Psi(v)$;

a threshold parameter $\tau > 0$;

an accuracy parameter $\epsilon > 0$;

a fixed barrier update parameter $\theta, 0 < \theta < 1$;

begin

$x := x^0; y := y^0; \mu := \mu^0$;

while $n\mu \geq \epsilon$ **do**

$\mu := (1-\theta)\mu$; $v := \sqrt{\frac{xy}{\mu}}$;

while $\Psi(v) > \tau$ **do**

calculate $(\Delta x, \Delta y)$ from (3.8);

calculate α the step size, then put

$x := x + \alpha\Delta x$; $y := y + \alpha\Delta y$; $v := \sqrt{\frac{xy}{\mu}}$;

end while

end while

end.

The parameters τ and α must be chosen in such a way that the number of iterations is as small as possible. This number depends on the choice of the proximity function ψ.

Recently, researchers are trying to improve the number of iterations, i.e. the algorithmic complexity, based on new kernel functions for large-update methods. The choice of the barrier functions plays an important role not only for the analysis but for the performance of the algorithm. Regarding the small-update methods, the results are the same for all functions, namely $O\left((1+2\kappa)\sqrt{n}\log\frac{n}{\epsilon}\right)$ which is the best complexity until now for this type of methods. The table 3.1 shows some recent kernel functions with the complexity with large-update methods.

3.1 Introduction of the kernel function

Our aim is to investigate a new kernel function with a trigonometric barrier term. This function is defined as follows :

$$\psi(t) = p\left(\frac{t^2-1}{2}\right) + \frac{4}{\pi}\left(\tan^p h(t) - 1\right), \; t > 0, p \geq \sqrt{2} \tag{3.9}$$

Kernel functions ψ_i	Large-update methods
$\frac{t^2-1}{2}-\left(\sqrt{3}-1\right)^p\int\limits_1^t\frac{dx}{\left(\tan\left(\frac{1+x}{4+2x}\pi\right)-1\right)^p}, p\geq 2$	$O\left(\left(1+2\kappa\right)\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$ [33]
$\frac{t^2-1}{2}+\frac{t^{1-q}-1}{q(q-1)}-\frac{q-1}{q}\left(t-1\right), q>0$	$O\left(\left(1+2\kappa\right)qn^{\frac{q+1}{2q}}\log\frac{n}{\epsilon}\right)$ [23]
$\frac{e\left(t^2-1\right)}{2}+\frac{e^{p(g_1(t)-e)-1}}{pr}, g_1\left(t\right)=e^{t^{-r}}, p\geq 1, r\geq 1$	$O\left(\left(1+2\kappa\right)\sqrt{n}\log n\log\frac{n\mu^0}{\epsilon}\right)$ [39]
$\frac{t^2-1}{2}+\frac{e^{p(g_2(t)-1)-1}}{pr}, g_2\left(t\right)=e^{t^{-r}-1}, p\geq 1, r\geq 1$	$O((1+2\kappa)\sqrt{n}log\frac{n\mu^0}{\epsilon})$ [39]
$\frac{t^2-1}{2}+\frac{4}{p\pi}\left(\tan^p\left(\frac{\pi}{2t+2}\right)-1\right), t\geq 1, p\geq 2$	$O(p(1+2\kappa)n^{\frac{2+p}{2(1+p)}}\log\frac{n}{\epsilon})$ [24]
$\frac{t^2-1}{2}-\int\limits_1^t\frac{1}{x^2p}e^{\frac{x}{p}-p}dx, p\geq 1, t>0$	$O\left(\left(1+2\kappa\right)p\sqrt{n}\left(\frac{1}{p}\log n+1\right)^2\log\frac{n}{\epsilon}\right)$ [33]
$\frac{p\left(t^2-1\right)}{2}-\int\limits_1^t pe^{p\left(\frac{1}{x}-1\right)}dx, p\geq 1$	$O\left(\sqrt{p^5n}\left(\log pn\right)^2\log\frac{n}{\epsilon}\right)$ [53]
$\frac{t^2-1-\log(t)}{2}+\frac{e^{\frac{1}{tq}}-1}{2q}, t>0, q\geq 1$	$O\left(q\sqrt{n}\left(\log\sqrt{n}\right)^{\frac{q+1}{q}}\log\frac{n}{\epsilon}\right)$ [22]

Table 3.1: Some kernel functions and their complexity with large-update methods

with $h\left(t\right)=\dfrac{\pi}{2t+2}$.

3.1.1 Some technical results

The first three derivatives of ψ are given by

$$\begin{aligned}
\psi'\left(t\right) &= pt+\frac{4}{\pi}p\left(1+\left(\tan h\left(t\right)\right)^2\right)\left(\left(\tan h\left(t\right)\right)^{p-1}h'\left(t\right)\right),\\
\psi''\left(t\right) &= p+\frac{4}{\pi}p\left(1+\left(\tan h\left(t\right)\right)^2\right)\left[\left[\left(p-1\right)\tan^{p-2}h\left(t\right)+\left(p+1\right)\tan^p h\left(t\right)\right]h'\left(t\right)^2\right.\\
&\quad \left.+\tan^{p-1}h\left(t\right)h''\left(t\right)\right],\\
\psi'''\left(t\right) &= \frac{4}{\pi}p\left(1+\tan^2 h\left(t\right)\right)\left[\left[\left(p-1\right)\left(p-2\right)\tan^{p-3}h\left(t\right)+2p^2\tan^{p-1}h\left(t\right)+\right.\right.\\
&\quad \left.\left(p+1\right)\left(p+2\right)\tan^{p+1}h\left(t\right)\right]h'\left(t\right)^3+\\
&\quad \left.3\left(p-1\right)\tan^{p-2}h\left(t\right)+3\left(p+1\right)\tan^p h\left(t\right)\right]h''\left(t\right)h'\left(t\right)+\\
&\quad \left.\tan^{p-1}h\left(t\right)h'''\left(t\right)\right].
\end{aligned}$$

with

$$h'\left(t\right)=-\frac{\pi}{2\left(t+1\right)^2},\ h''\left(t\right)=\frac{\pi}{\left(t+1\right)^3},\ h'''\left(t\right)=\frac{-3\pi}{\left(t+1\right)^4},$$

are the first three derivatives of h. The next lemma serves to prove that the new kernel function (3.9) is eligible.

Lemma 3.1.1. *Let ψ be as defined in (3.9), and $t>0$. Then*

$$\psi''\left(t\right)>1, \qquad \text{(a)}$$

$$t\psi''(t)+\psi'(t)>0, \tag{b}$$

$$t\psi''(t)-\psi'(t)>0, \tag{c}$$

$$\psi'''(t)<0\,, \tag{d}$$

$$2\psi''(t)^2-\psi'''(t)\psi'(t)>0, t<1. \tag{e}$$

Proof. $a)$ we have :

$$0<h(t)<\frac{\pi}{2}, \forall t>0 \tag{*}$$

because h is decreasing function in $]0,+\infty[$ and

$$\lim_{t\to 0^+} h(t)=\frac{\pi}{2}, \lim_{t\to+\infty} h(t)=0,$$

we get that

$$\tan h(t)>0, t>0, \tag{**}$$

then, it comes that

$$\left[(p-1)\tan^{p-2}h(t)+(p+1)\tan^p h(t)\right]h'(t)^2+\tan^{p-1}h(t)\,h''(t)>0 \tag{***}$$

which implies that

$$\psi''(t)>1.$$

$b)$ Using (**) and (***) we have

$$t\psi''(t)-\psi'(t)=\frac{4p}{\pi}\left(1+\tan^2 h(t)\right)\left[\begin{array}{c} t\left[(p-1)\tan^{p-2}h(t)+(p+1)\tan^p h(t)\right]h'(t)^2 \\ +t\tan^{p-1}h(t)\,h''(t) \\ +\tan^{p-1}h(t)\,(-h'(t)) \end{array}\right]>0, t>0.$$

$c)$

$$t\psi''(t)+\psi'(t)=2pt+\frac{p}{\pi}\left(1+\tan^2 h(t)\right)\left[\begin{array}{c} t(p-1)(h'(t))^2\tan^{p-2}h(t)+ \\ \tan^{p-1}h(t)\left[t(h'(t))^2(p+1)+th''(t)+h'(t)\right] \end{array}\right]$$

and

$$t\left(h'\left(t\right)\right)^{2}\left(p+1\right)+th''\left(t\right)+h'\left(t\right)=\frac{4\pi}{\left(2t+2\right)^{4}}\left[2t^{2}+\pi\left(p+1\right)t\left(\tan h\left(t\right)-\frac{2}{\left(p+1\right)\pi t}\right)\right].$$

Now, define

$$g\left(t\right)=\tan h\left(t\right)-\frac{2}{\left(p+1\right)\pi t},$$

with simple calculations we find that the derivative of g is negative. Thus $g\left(t\right)$ is decreasing in $]0,+\infty[$, and since $\lim_{t\rightarrow+\infty}g\left(t\right)=0$, then

$$\tan h\left(t\right)-\frac{2}{\left(p+1\right)\pi t}>0,t>0,$$

which leads to

$$t\psi''\left(t\right)+\psi'\left(t\right)>0.$$

d) Using (**), $h'\left(t\right)<0$ and $h'''\left(t\right)<0$, then $\psi'''\left(t\right)<0$.

e) We put

$$\begin{aligned}\psi'\left(t\right) &= pt+A\left(t\right)h'\left(t\right)\tan^{p-1}h\left(t\right)\\ \psi''\left(t\right) &= p+A\left(t\right)B\left(t\right)\\ \psi'''\left(t\right) &= A\left(t\right)C\left(t\right)\end{aligned}$$

with

$$\begin{aligned}A\left(t\right) &= \frac{4}{\pi}p\left(1+\tan^{2}h\left(t\right)\right)\\ B\left(t\right) &= \left[\left(p-1\right)\tan^{p-2}h\left(t\right)+\left(p+1\right)\tan^{p}h\left(t\right)\right]h'\left(t\right)^{2}+\tan^{p-1}h\left(t\right)h''\left(t\right)\\ C\left(t\right) &= [[\left(p-1\right)\left(p-2\right)\tan^{p-3}h\left(t\right)+2p^{2}\tan^{p-1}h\left(t\right)+\\ &\quad \left(p+1\right)\left(p+2\right)\tan^{p+1}h\left(t\right)]\left(h'\left(t\right)\right)^{3}+\\ &\quad \left[3\left(p-1\right)\tan^{p-2}h\left(t\right)+3\left(p+1\right)\tan^{p}h\left(t\right)\right]h''\left(t\right)h'\left(t\right)+\\ &\quad \tan^{p-1}h\left(t\right)h'''\left(t\right)]\end{aligned}$$

then

$$2\psi''\left(t\right)^{2}-\psi'''\left(t\right)\psi'\left(t\right)=2p^{2}+A\left(t\right)^{2}\left[2B\left(t\right)^{2}-C\left(t\right)h'\left(t\right)\tan^{p-1}h\left(t\right)\right]+A\left(t\right)4pB\left(t\right)-ptA\left(t\right)C\left(t\right),$$

using (**), (***) and (d), we have $A(t) > 0, B(t) > 0$ and $C(t) < 0$, then

$$A(t) 4pB(t) - ptA(t) C(t) > 0, t > 0,$$

it remains to see if $2B(t)^2 - C(t) h'(t) \tan^{p-1} h(t)$ is positive,

$$\begin{aligned} 2B(t)^2 - C(t) h'(t) \tan^{p-1} h(t) &= h'(t)^4 \left[p(p-1) \tan^{2(p-2)} h(t) + \left(2p^2 - 4\right) \tan^{2p-2} h(t)\right. \\ &\quad \left. + p(p+1) \tan^{2p} h(t)\right] \\ &\quad + h'(t)^2 h'' \left[(p-1) \tan^{2p-3} h(t) + (p+1) \tan^{2p-1} h(t)\right] \\ &\quad + \tan^{2(p-1)} h(t) \left(2h''(t)^2 - h'(t) h'''(t)\right) \end{aligned}$$

from (**) and definitions of $h'(t), h''(t)$ and $h'''(t)$ and

$$2h''(t)^2 - h'(t) h'''(t) = \frac{\pi^2}{2(t+1)^6} > 0, t > 0,$$

it follows that

$$2B(t)^2 - C(t) h'(t) \tan^{p-1} h(t) > 0,$$

which implies that

$$2\psi''(t)^2 - \psi'''(t) \psi'(t) > 0, t < 1.$$

□

It follows that $\psi(1) = \psi'(1) = 0$ and $\psi''(t) \geq 0$, proving that ψ is defined by $\psi''(t)$,

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\zeta) \, d\zeta d\xi. \tag{3.10}$$

Property (b) is equivalent to the convexity of the composed function $z \to \psi(e^z)$ and this is true if and only if

$$\psi\left(\sqrt{t_1 t_2}\right) \leq \frac{1}{2} \left(\psi(t_1) + \psi(t_2)\right), \forall t_1, t_2 \geq 0.$$

We say that ψ is exponentially convex, or shortly, e-convex, when $t > 0$.

Lemma 3.1.2. *Let ψ be as defined in (3.9), one has*

$$\psi(t) < \frac{1}{2} \psi''(t) (t-1)^2, \text{ if } t > 1.$$

Proof. By Taylor's theorem and $\psi(1) = \psi'(t) = 0$, we obtain

$$\psi(t) < \frac{1}{2}\psi''(t)(t-1)^2 + \frac{1}{6}\psi'''(\xi)(\xi-1)^3$$

where $1 < \xi < t$, if $t > 1$. Since $\psi'''(t) < 0$, then

$$\psi(t) < \frac{1}{2}\psi''(t)(t-1)^2,$$

the lemma follows. □

Lemma 3.1.3. *Let ψ be as defined in (3.9), one has*

$$t\psi'(t) \geq \psi(t) \text{ if } t \geq 1.$$

Proof. Defining $g(t) = t\psi'(t) - \psi(t)$, one has $g(1) = 0$ and $g'(t) = t\psi''(t) \geq 0$. Hence $g(t) \geq 0$, and the lemma follows. □

3.1.2 Proximity measure

We now introduce a norm-based proximity measure $\delta(v)$

$$\delta(v) = \frac{1}{2}\|\nabla\Psi(v)\| = \frac{1}{2}\sqrt{\sum_{i=1}^{n}\psi'(v_i)^2} = \frac{1}{2}\|d_x + d_y\|. \tag{3.11}$$

Since $\Psi(v)$ is strictly convex and attains its minimal value zero at $v = e$, we have

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

For the analysis of the algorithm we need to establish relations between $\Psi(v)$ and $\delta(v)$. A crucial observation is that the inverse function of $\psi(t)$, for $t \geq 1$, plays an important role in this relation. Let $\rho : [0, \infty[\to]0, 1]$ denotes the inverse function of the restriction of $\frac{-1}{2}\psi'(t)$ for $0 < t \leq 1$, and $\varrho : [0, \infty[\to [1, \infty[$ denotes the inverse function of $\psi(t)$ for $t \geq 1$, then

$$y = \psi(t) \Leftrightarrow t = \varrho(y), t \geq 1, \tag{3.12}$$

$$y = \frac{-1}{2}\psi'(t) \Longleftrightarrow t = \rho(y), t \leq 1. \tag{3.13}$$

The following theorem gives a lower bound of $\delta(v)$ as a function of $\Psi(v)$.

Theorem 3.1.1. *(Theorem 5.1 of [40]) Let ϱ be as defined in (3.12). One has*

$$\delta(v) \geq \frac{1}{2}\psi'(\varrho(\Psi(v))).$$

Corollary 3.1.1. *(Corollary 5.2 of [40]) Thus we have*

$$\delta(v) \geq \frac{\Psi(v)}{2\varrho(\Psi(v))}.$$

Theorem 3.1.2. *If $\Psi(v) \geq 1$ then*

$$\delta(v) \geq \frac{1}{6}\Psi^{1/2}. \tag{3.14}$$

Proof. The inverse function of $\psi(t)$ for $t \in [1, \infty[$ is obtained by

$$\psi(t) = p\left(\frac{t^2-1}{2}\right) + \frac{4}{\pi}\left(\tan^p\left(\frac{\pi}{2t+2}\right) - 1\right) = y,\ t > 1.$$

We derive an upper bound of t. From (3.10) and $\psi''(t) \geq 1$,

$$y = \psi(t) = \int_1^t \int_1^\xi \psi''(\zeta)\, d\zeta d\xi. \geq \int_1^t \int_1^\xi d\zeta d\xi = \frac{(t-1)^2}{2},$$

which implies

$$t = \varrho(y) \leq 1 + \sqrt{2y}. \tag{3.15}$$

Assuming that $y \geq 1$, we get $t = \varrho(y) \leq \sqrt{y} + \sqrt{2y} \leq 3y^{1/2}$, assuming also $\Psi(v) \geq 1$, we get $\varrho(\Psi(v)) \leq 3\Psi(v)^{1/2}$. Then by Corollary 3.1.1, we have

$$\delta(v) \geq \frac{\Psi(v)}{2\varrho(\Psi(v))} \geq \frac{1}{6}\Psi^{1/2},$$

which finishes the proof. □

Remark 3.1.1. *If $\Psi(v) \geq 1$, by (3.14) we have*

$$\delta(v) \geq \frac{1}{6}. \tag{3.16}$$

3.1.3 Growth behavior of the barrier function

During the algorithm, the largest values of $\Psi(v)$ occur just after the updates of μ. In this section, we obtain an estimate of the effect of an update of μ on the value of $\Psi(v)$. We begin with an important theorem that is valid for all kernel functions $\psi(t)$ that are strictly convex (a), and satisfy (c).

Theorem 3.1.3. *(Theorem 5.1 of [40]) Let* $\varrho : [0,\infty[\to [1,\infty[$ *the inverse function of* ψ *in* $[0,\infty[$. *Then for all positive vector* v *and* $\beta > 1$ *we have*

$$\Psi(\beta v) \leq n\psi\left(\beta\varrho\left(\frac{\Psi(v)}{n}\right)\right). \tag{3.17}$$

Corollary 3.1.2. *(Corollary 5.2 of [40]) Let* $1 < \theta < 0$ *and* $v_+ = \frac{v}{\sqrt{1-\theta}}$. *Then*

$$\Psi(v_+) \leq n\psi\left(\frac{\varrho\left(\frac{\Psi(v)}{n}\right)}{\sqrt{1-\theta}}\right). \tag{3.18}$$

Remark 3.1.2. *Assuming that the update parameter* θ *and the threshold value* τ *are given, at the beginning of each outer iteration, we have* $\Psi(v) \leq \tau$. *We define*

$$L = L(n,\theta,\tau) = n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right). \tag{3.19}$$

L *is an upper bound of* $\Psi(v_+)$ *which is the value of* $\Psi(v)$ *after the* μ*-update.*

3.2 Analysis of the algorithm

After a step with size α the new iterates are

$$\begin{aligned} x_+ &= x + \alpha\Delta x = \frac{x}{v}(v + \alpha d_x), \\ y_+ &= y + \alpha\Delta y = \frac{y}{v}(v + \alpha d_y). \end{aligned}$$

Recall that during an inner iteration the parameter μ is fixed. Hence, after the step the new v-vector is given by

$$v_+ = \sqrt{\frac{x_+ y_+}{\mu}} = \sqrt{(v + \alpha d_x)(v + \alpha d_y)}$$

then

$$v_+^2 = \frac{x_+ y_+}{\mu} = (v + \alpha d_x)(v + \alpha d_y). \tag{3.20}$$

In the sequel we use the following notations

$$\nu = \min_{i\in I_+} v_i, \delta = \delta(v), \sigma_+ = \sum_{i\in I_+} d_{x_i} d_{y_i}, \sigma_- = -\sum_{i\in I_-} d_{x_i} d_{y_i}. \tag{3.21}$$

Since $\{M, N\}$ is $P_*(\kappa) - pair$ and $-M\Delta x + N\Delta y = 0$, for $(\Delta x, \Delta y) \in \mathbb{R}^n \times \mathbb{R}^n$ then

$$(1+4\kappa)\sum_{i\in I_+}\Delta x_i\Delta y_i + \sum_{i\in I_-}\Delta x_i\Delta y_i \geq 0,$$

where $I_+ = \{i \in I : \Delta x_i \Delta y_i \geq 0\}, I_- = I - I_+$.
Because $d_x d_y = \frac{v^2\Delta x\Delta y}{xy} = \frac{\Delta x\Delta y}{\mu}$ and $\mu > 0$, we get

$$(1+4\kappa)\sum_{i\in I_+} d_{x_i}d_{y_i} + \sum_{i\in I_-} d_{x_i}d_{y_i} \geq 0 = (1+4\kappa)\,\sigma_+ - \sigma_- \geq 0. \tag{3.22}$$

The following lemma gives an upper bound of σ_+ and σ_-.

Lemma 3.2.1. *One has $\sigma_+ \leq \delta^2$ and $\sigma_- \leq (1+4\kappa)\,\delta^2$.*

Proof. By definition of σ_+ and σ_-, we have

$$\sigma_+ = \sum_{i\in I_+} d_{x_i}d_{y_i} \leq \frac{1}{4}\sum_{i\in I_+}(d_{x_i}+d_{y_i})^2 \leq \frac{1}{4}\sum_{i\in I_+}(d_{x_i}+d_{y_i})^2$$

$$= \frac{1}{4}\|d_x + d_y\|^2 = \delta^2.$$

Using (3.22), we get

$$(1+4\kappa)\,\sigma_+ - \sigma_- \geq 0.$$

Then

$$\sigma_- \leq (1+4\kappa)\,\sigma_+ \leq (1+4\kappa)\,\delta^2.$$

Which finishes the proof. □

The following lemma gives an upper bound of $\|d_x\|$ and $\|d_y\|$.

Lemma 3.2.2. *One has*

$$\sum_{i\in I_+}\left(d_{x_i}^2 + d_{y_i}^2\right) \leq 4\,(1+2\kappa)\,\delta^2,$$

$$\|d_x\| \leq 2\sqrt{1+2\kappa}\delta$$

and

$$\|d_y\| \leq 2\sqrt{1+2\kappa}\delta.$$

Proof. According to definitions (3.11) and (3.21), we have

$$\delta = \frac{1}{2}\|d_x + d_y\| \text{ and } \sum_{i\in I} d_{x_i}d_{y_i} = \sigma_+ - \sigma_-.$$

Then

$$\begin{aligned} 2\delta &= \|d_x + d_y\| = \sqrt{\sum_{i=1}^{n} \left(d_{x_i} + d_{y_i}\right)^2} \\ &= \sqrt{\sum_{i=1}^{n} \left(d_{x_i}^2 + d_{y_i}^2\right) + 2\left(\sigma_+ - \sigma_-\right)}. \end{aligned}$$

Using (3.22) and Lemma 3.2.1 we get

$$\begin{aligned} 2\delta &\geq \sqrt{\sum_{i=1}^{n} \left(d_{x_i}^2 + d_{y_i}^2\right) + 2\left(\frac{1}{1+4\kappa}\sigma_+ - \sigma_-\right)} \\ &= \sqrt{\sum_{i=1}^{n} \left(d_{x_i}^2 + d_{y_i}^2\right) - \frac{8\kappa}{1+4\kappa}\sigma_-}. \end{aligned}$$

Then we get

$$4\delta^2 + \frac{8\kappa}{1+4\kappa}\sigma_- \geq \sum_{i=1}^{n} \left(d_{x_i}^2 + d_{y_i}^2\right),$$

by using Lemma 3.2.1 again, we have

$$\|d_x\| \leq \sqrt{\sum_{i=1}^{n} \left(d_{x_i}^2 + d_{y_i}^2\right)} \leq 2\sqrt{1+2\kappa}\delta.$$

Similarly, we can prove that

$$\|d_y\| \leq 2\sqrt{1+2\kappa}\delta.$$

So the lemma follows. □

Our goal is to find an upper bound for

$$f(\alpha) = \Psi(v_+) - \Psi(v) = \Psi\left(\sqrt{(v+\alpha d_x)(v+\alpha d_y)}\right) - \Psi(v),$$

where $\Psi : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i). \tag{3.23}$$

It is clear that $f(\alpha)$ is not necessarily convex in α. To simplify the analysis we use a convex upper

bound for $f(\alpha)$. Such a bound is obtained by using that $\psi(t)$ satisfies the condition (b). This implies

$$\begin{aligned}\Psi(v_+) &= \Psi\left(\sqrt{(v+\alpha d_x)(v+\alpha d_y)}\right)\\ &\leq \frac{1}{2}\left[\Psi(v+\alpha d_x)+\Psi(v+\alpha d_y)\right].\end{aligned}$$

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

$$f_1(\alpha)=\frac{1}{2}\left[\Psi(v+\alpha d_x)+\Psi(v+\alpha d_y)\right]-\Psi(v).$$

$f_1(\alpha)$ is a convex function in α, because $\Psi(v)$ is convex. Obviously, $f(0)=f_1(0)=0$. The derivative of $f_1(\alpha)$ in α

$$f_1'(\alpha)=\frac{1}{2}\sum_{i=1}^{n}\left(\psi'(v_i+\alpha d_{x_i})d_{x_i}+\psi'(v_i+\alpha d_{y_i})d_{y_i}\right).$$

This gives, using the equation $d_x+d_y=-\nabla\Psi(v)$ and (3.10),

$$f_1'(0)=\frac{1}{2}\nabla\Psi(v)^t(d_x+d_y)=-\frac{1}{2}\nabla\Psi(v)^t\nabla\Psi(v)=-2\delta(v)^2,$$

$$f_1''(\alpha)=\frac{1}{2}\sum_{i=1}^{n}\left(\psi''(v_i+\alpha d_{x_i})d_{x_i}^2+\psi''(v_i+\alpha d_{y_i})d_{y_i}^2\right).$$

The next lemma gives an upper bound of $f_1(\alpha)$ in terms of δ and $\psi''(t)$.

Lemma 3.2.3. *(Lemma 5.4 of [40]) One has*

$$f_1''(\alpha)\leq 2(1+2\kappa)\delta^2\psi''\left(\nu-2\alpha\sqrt{1+2\kappa}\delta\right),$$

since $f_1(\alpha)$ is convex, we will have $f'(\alpha)\leq 0$ for any α less than or equal to the value where $f_1(\alpha)$ is minimal, and vice versa.

Lemma 3.2.4. *(Lemma 5.5 f [40]) we have $f_1'(\alpha)\leq 0$ if α satisfies*

$$-\psi'\left(\nu-2\alpha\delta\sqrt{1+2\kappa}\right)+\psi'(\nu)\leq\frac{2\delta}{(1+2\kappa)}. \tag{3.24}$$

Lemma 3.2.5. *(Lemma 5.6 [40]) The largest value of the step α which satisfies (3.24) is given by*

$$\bar{\alpha}=\frac{1}{2\delta\sqrt{1+2\kappa}}\left[\rho(\delta)-\rho\left(1+\frac{1}{\sqrt{1+2\kappa}}\delta\right)\right],$$

furthermore

$$\bar{\alpha} \geq \frac{1}{(1+2\kappa)\,\psi''\left(\rho\left(1+\frac{1}{\sqrt{1+2\kappa}}\delta\right)\right)}. \tag{3.25}$$

We define

$$\tilde{\alpha} = \frac{1}{(1+2\kappa)\,\psi''\left(\rho\left(1+\frac{1}{\sqrt{1+2\kappa}}\delta\right)\right)} \tag{3.26}$$

as a default step size. From Lemma 3.2.5, α satisfies (3.26). From (3.25) we have $\bar{\alpha} \geq \tilde{\alpha}$.

Lemma 3.2.6. *(Lemma 5.9 of [40]) Let $h(t)$ be a convex function twice differentiable with $h(0) = 0, h'(0) < 0$ and $h(t)$ reaches its (global) minimum at $t^* > 0$. If $h''(t)$ is increasing for $t \in [0, t^*]$ then*

$$h(t) = \frac{th'(0)}{2}, 0 \leq t \leq t^*.$$

Lemma 3.2.7. *(Lemma 5.10 of [40]) If α satisfies (3.25) then*

$$f(\alpha) \leq -\alpha\delta^2. \tag{3.27}$$

Theorem 3.2.1. *Let ρ be defined in (3.13) and α in (3.25). Then*

$$\begin{aligned} f(\bar{\alpha}) &\leq -\frac{\delta^2}{(1+2\kappa)\,\psi''\left(\rho\left(1+\frac{1}{\sqrt{1+2\kappa}}\delta\right)\right)} \\ &\leq -\frac{\psi^{\frac{p}{2(p+1)}}}{6\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12p\right)^2(1+2\kappa)}. \end{aligned}$$

Proof. First,

$$f(\bar{\alpha}) \leq -\frac{\delta^2}{(1+2k)\,\psi''\left(\rho\left(1+\frac{1}{\sqrt{1+2\kappa}}\delta\right)\right)}$$

is evident. To get the inverse function $t = \rho(y)$ of $\frac{-1}{2}\psi'(t)$ such that $t \in (0, 1]$, we need to solve the equation

$$\begin{aligned} 2y &= -\left[pt + \frac{4ph'(t)}{\pi}\left(1+\tan^2 h(t)\right)\tan^{p-1} h(t)\right] \\ &= -pt - \frac{4ph'(t)}{\pi}\sec^2 h(t)\tan^{p-1} h(t) \\ &= -pt - \frac{4ph'(t)}{\pi}\csc^2 h(t)\tan^{p+1} h(t) \end{aligned}$$

which implies

$$\begin{aligned}\csc^2 h(t) \cdot \tan^{p+1} h(t) &= (2y+pt)\left(-\frac{\pi}{4ph'(t)}\right) \\ &= (2y+pt)\left(\frac{(t+1)^2}{2p}\right).\end{aligned}$$

For $t \leq 1$, we have

$$\frac{(t+1)^2}{2p}(2y+pt) \leq \frac{2}{p}(2y+p),$$

which implies

$$\csc^2 h(t) \tan^{p+1} h(t) \leq \frac{2}{p}(2y+p).$$

Let

$$t = \rho\left(1 + \frac{1}{\sqrt{1+2\kappa}}\delta\right),$$

which is equivalent to

$$2\left(1 + \frac{1}{\sqrt{1+2\kappa}}\delta\right) = -\psi'(t),$$

because $1 + \frac{1}{\sqrt{1+2\kappa}} \leq 2$, for $\kappa \geq 0$, and $\sin^2 h(t) \leq 1$, one has

$$\tan h(t) \leq \left(\frac{2}{p}(4\delta+p)\right)^{\frac{1}{p+1}}.$$

Since : $sec^2 h(t) = 1 + \tan^2 h(t)$, then

$$\begin{aligned}\tan^2 h(t) &\leq \left(\frac{2}{p}(4\delta+p)\right)^{\frac{2}{p+1}}, \quad \tan^{p-1} h(t) \leq \left(\frac{2}{p}(4\delta+p)\right)^{\frac{p-1}{p+1}}, \\ \tan^{p-2} h(t) &\leq \left(\frac{2}{p}(4\delta+p)\right)^{\frac{p-2}{p+1}}, \quad \tan^{p} h(t) \leq \left(\frac{2}{p}(4\delta+p)\right)^{\frac{p}{p+1}},\end{aligned}$$

and

$$h''(t) = \frac{\pi}{(t+1)^3} \leq \pi, \quad (h'(t))^2 = \left(\frac{-2\pi}{4(t+1)}\right)^2 \leq \frac{\pi^2}{4}, \ \ \forall t \in (0,1],$$

and using $\left(\frac{8}{p}\delta+2\right)\geq 1$, we get

$$\begin{aligned}\psi''(t) &\leq p+\frac{4p}{\pi}\left(1+\left(\frac{8}{p}\delta+2\right)^{\frac{2}{p+1}}\right)\left(\begin{array}{c}\left(\begin{array}{c}(p-1)\left(\frac{8}{p}\delta+2\right)^{\frac{p-2}{p+1}}+\\ (p+1)\left(\frac{8}{p}\delta+2\right)^{\frac{p}{p+1}}\end{array}\right)\frac{\pi^2}{4}+\\ \pi\left(\frac{8}{p}\delta+2\right)^{\frac{p-1}{p+1}}\end{array}\right)\\ &\leq \left(p+\frac{4\pi}{p}(2)+\pi\right)\left(\frac{8}{p}\delta+2\right)^{\frac{p+2}{p+1}}\\ &\leq \left(9p+4\pi p^2\right)\left(\frac{8}{p}\delta+2\right)^{\frac{p+2}{p+1}}.\end{aligned}$$

So from (3.25) we have

$$\bar{\alpha}\geq\frac{1}{(1+2\kappa)\left(9p+4p^2\pi\right)\left(\frac{8}{p}\delta+2\right)^{\frac{p+2}{p+1}}},$$

using $\delta\geq\frac{1}{6}$, we get

$$\bar{\alpha}\geq\frac{1}{(1+2\kappa)\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^{\frac{p+2}{p+1}}\delta^{\frac{p+2}{p+1}}},$$

then

$$f(\bar{\alpha})\leq-\frac{\delta^2}{(1+2\kappa)\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^{\frac{p+2}{p+1}}\delta^{\frac{p+2}{p+1}}}\leq-\frac{\delta^{\frac{p}{p+1}}}{(1+2\kappa)\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^2},$$

and since $\delta\geq\frac{1}{6}\psi^{\frac{1}{2}}$, we get

$$\begin{aligned}f(\bar{\alpha}) &\leq -\left(\frac{1}{6}\psi^{\frac{1}{2}}\right)^{\frac{p}{p+1}}\cdot\frac{1}{(1+2\kappa)\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^2}\\ &\leq -\frac{\psi^{\frac{p}{2(p+1)}}}{6\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^2(1+2\kappa)}.\end{aligned}$$

which ends the proof. □

3.2.1 Algorithmic complexity

Let K be the number of inner iterations. An upper bound for the total number of iterations is obtained by multiplying K by the number of the barrier parameter updates, which is bounded by $\frac{1}{\theta}\log\frac{n}{\epsilon}$.

Lemma 3.2.8. *(Lemma 6.1 of [40]) Let $t_0, t_1, ..., t_k$, a sequence of positive numbers that satisfies*

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

where $\beta > 0$ and $0 < \gamma \leq 1$, then : $K \leq \left\lfloor \frac{t_0^{\gamma}}{\beta\gamma} \right\rfloor$.

Lemma 3.2.9. *One has*

$$K \leq 6\left(9p + 4\pi p^2\right)\left(\frac{8}{p} + 12\right)^2 (1 + 2\kappa)\, \Psi_0^{\frac{2+p}{2(1+p)}}.$$

Proof. The definition of K implies that $\Psi_{K-1} > \tau$ and $\Psi_K \leq \tau$ and $\Psi_{k+1} \leq \Psi_k - \beta\left(\Psi_k\right)^{1-\gamma}, k = 0, 1, 2, ..., K-1$, with $\beta = \frac{1}{6(9p+4\pi p^2)\left(\frac{8}{p}+12\right)^2(1+2\kappa)}$ and $\gamma = \frac{2+p}{2(1+p)}$. Applying the previous lemma with $t_k = \Psi_k$, gives the desired inequality. □

Using $\psi_0 \leq L$, where the number L is given by (3.19) and the previous lemma we obtain the next upper bound on the total number of iterations:

$$\frac{6\left(9p + 4\pi p^2\right)\left(\frac{8}{p} + 12\right)^2 (1 + 2\kappa)\, L^{\frac{2+p}{2(1+p)}}}{\theta} \log \frac{n}{\epsilon}. \tag{3.28}$$

Large-Update methods

We have just established that (3.28) is an upper bound for the total number of iterations, using (3.9) and $t = \varrho(y) \leq 1 - \sqrt{2y}$, by substitution in (3.19) we obtain

$$\begin{aligned} L &\leq n\frac{\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right)^2 - 1}{2} \\ &\leq \frac{n}{2(1-\theta)}\left(\theta + 2\sqrt{2\frac{\tau}{n}} + \frac{2\tau}{n}\right) = \frac{\left(\theta + 2\sqrt{2\tau n} + 2\tau\right)}{2(1-\theta)}. \end{aligned}$$

Using (3.28), the total number of iterations is bounded by

$$\frac{K}{\theta}\log\frac{n}{\epsilon} \leq \frac{6\left(9p + 4\pi p^2\right)\left(\frac{8}{p} + 12\right)^2 (1 + 2k)}{\theta\left(2(1-\theta)^{\frac{2+p}{2(1+p)}}\right)}\left(\theta n + 2\sqrt{2\tau n} + 2\tau\right)^{\frac{2+p}{2(1+p)}} \log\frac{n}{\epsilon}.$$

For large-update methods we use $\tau = O(n)$ and $\theta = O(1)$. The algorithmic complexity is then of order $O\left(p^2(1 + 2\kappa)\, n^{\frac{2+p}{2(1+p)}} \log\frac{n}{\epsilon}\right)$.

Remark 3.2.1. *With a special choice of the parameter p $\left(p = O\left((\log n)^{\frac{1}{2}}\right)\right)$ the algorithm has*

$O\left((1+2\kappa)\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$ *iteration complexity. It is the best known iteration bound for* $P_*(\kappa)$-*HLCP.*

Small-Update Methods

We have $\tau = O(1)$ and $\theta = O\left(\frac{1}{\sqrt{n}}\right)$. Using

$$\psi(t) < \frac{1}{2}\psi''(1)(t-1)^2, t > 1,$$

with

$$\psi''(1) = 2p + \frac{\pi}{4}p^2,$$

we get

$$\begin{aligned} L &\leq \frac{n}{2}\left(2p+\frac{\pi}{4}p^2\right)\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}-1\right)^2 \\ &\leq \frac{n\left(8p+\pi p^2\right)}{8(1-\theta)}\left(\frac{1+\sqrt{2\frac{\tau}{n}-\sqrt{1-\theta}}}{\sqrt{1-\theta}}\right)^2. \end{aligned}$$

Using $1-\sqrt{1-\theta} = \frac{\theta}{1+\sqrt{1-\theta}} \leq \theta$, we find

$$L \leq \frac{n\left(8p+\pi p^2\right)}{8(1-\theta)}\left(\theta\sqrt{n}+\sqrt{2\tau}\right)^2.$$

We conclude that the total number of iterations is bounded by

$$\frac{K}{\theta}\log\frac{n}{\epsilon} \leq \frac{6\left(9p+4\pi p^2\right)\left(\frac{8}{p}+12\right)^2(1+2k)\left(8p+\pi p^2\right)^{\frac{2+p}{2(1+p)}}}{\theta\left(8(1-\theta)\right)^{\frac{2+p}{2(1+p)}}}\left(\theta\sqrt{n}+\sqrt{2\tau}\right)^{\frac{2+p}{1+p}}\log\frac{n}{\epsilon},$$

hence an algorithmic complexity of order $O\left((1+2\kappa)\sqrt{n}\log\frac{n}{\epsilon}\right)$.

3.3 Numerical results

In this section, we present some comparative numerical results that are derived by applying our algorithm with the new proposed kernel function and some kernel functions described in table (3.2). We have chosen the parameters of the algorithm for all the considered kernel functions as : $\epsilon = 10^{-6}, \tau = O(n)$ and $\theta = 0.9$. Note that for all experiments, the value of the step size in the inner iterations has been chosen by a dynamic procedure introduced in [50] and based on an approximate value of the

$\psi_i(t)$	kernel functions	$h(t)$	step-size
$\psi_1(t)$	$p\left(\frac{t^2-1}{2}\right)+\frac{4}{\pi}\left(\tan^p h(t)-1\right), t>0, p\geq\sqrt{2}$	$\frac{\pi}{2t+2}$	$\frac{1}{(1+2\kappa)(9p+4\pi p^2)\left(\frac{8}{p}+2\right)^{\frac{p+2}{p+1}}}$
$\psi_2(t)$ [51]	$\frac{t^2-1}{2}-\log t+\frac{1}{8}\tan^2 h(t), t>0$	$\frac{\pi(1-t)}{2+4t}$	$\frac{1}{(1+2\kappa)5020\delta^{\frac{4}{3}}}$
$\psi_3(t)$ [12]	$t-1+\frac{t^{1-q}-1}{q-1}, t>0, q\geq 2$	-	$\frac{1}{(1+2\kappa)q(2\delta+1)^{\frac{1}{q}}(4\delta+1)}$
$\psi_4(t)$ [3]	$\frac{t^2-1}{2}+\frac{q^{\frac{1}{t}-1}}{q\log q}-\frac{q-1}{q}(t-1), q>1$	-	$\frac{1}{(1+2\kappa)(\log q+2)(1+4\delta)\left(2+\frac{\log(1+4\delta)}{\log q}\right)}$
$\psi_5(t)$ [14]	$\frac{t^2-1}{2}+\frac{4}{p\pi}\left(\tan^p h(t)-1\right), t>0, p\geq 2$	$\frac{\pi}{2t+2}$	$\frac{1}{(1+2\kappa)(9+4p\pi)(8\delta+2)^{\frac{p+2}{p+1}}}$

Table 3.2: Some kernel functions

default step size in the related references. The dynamic step-size is defined as follows:

$$\alpha_d=\begin{cases}2\alpha, \|\Delta x\|\geq n\\ 5\alpha, 1<\|\Delta x\|<n\\ 10\alpha, \|\Delta x\|<1,\end{cases}$$

where α is the theoretical step-size. The number of inner iterations and outer iterations are denoted by "**Inn**" and "**Out**", respectively.

Problem 3.1. Consider an HLCP problem with the following setting:

$$M=\begin{pmatrix}1.5 & 10.5 & 1.05 & 1.05 & 2 & 0 & 0\\ 2 & 1.5 & 6 & 6 & 1.5 & 1.5 & 2\\ 1.5 & 3.5 & 55 & 9.5 & 100 & 1500 & -5\\ 5 & 1.5 & 3.5 & 1 & 0 & 3 & 40.5\\ 3 & 3 & 8.25 & 2.5 & 2.25 & 0 & 2.5\\ 1.5 & 5 & 5 & 0 & 0 & 1.05 & 0\\ 1.5 & 1.05 & 1 & 1.05 & 1 & 2 & 1000\end{pmatrix},$$

$$N=\begin{pmatrix}0.5 & 9.5 & 0.95 & 0.95 & 2 & 0 & 0\\ 2 & 0.5 & 6 & 6 & 0.5 & 0.5 & 2\\ 0.5 & 2.5 & 4.5 & 8.5 & 100 & 1500 & -5\\ 5 & 0.5 & 2.5 & 1 & 0 & 3 & 39.5\\ 3 & 3 & 7.75 & 1.5 & 1.75 & 0 & 1.5\\ 0.5 & 5 & 5 & 0 & 0 & 0.95 & 0\\ 0.5 & 0.95 & 1 & 0.95 & 1 & 2 & 1000\end{pmatrix},$$

$$q=(-16.4, 23, 3206, 79, 13, -1.1, 2004.8)^T.$$

The exact solution is $x^* = (0,0,2,2,2,2,2)^T$ and $y^* = (2,2,0,0,0,0,0)^T$. The numerical results are summarized in table 3.3.

	$\psi_1(t)$	$\psi_2(t)$	$\psi_3(t)$	$\psi_4(t)$	$\psi_5(t)$
Inn	5	282	10	16	5
Out	8	8	8	8	8
Time	0.07	0.18	0.062	0.06	0.06

Table 3.3: Numerical results of problem 3.1.

Problem 3.2. For $c \geq 0$, the $P_*(\kappa)$-HLCP is given by

$$M = \begin{pmatrix} 0 & 1+4\kappa & 0 \\ -1 & 0 & 0 \\ 0 & 0 & c \end{pmatrix}, N = I \text{ and } q = (0.01, 0.501, -0.49)^T .$$

The pair $\{M, N\}$ is a $P_*(\kappa)$ for all $\kappa \geq 0$ (see [19]). For example if $\kappa = 1$ and $c = 1$, then the HLCP (a non monotone HLCP) is given by

$$M = \begin{pmatrix} 0 & 5 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, N = I \text{ and } q = (0.01, 0.501, -0.49)^T .$$

The exact solution $x^* = (0,0,0.49)^T$ and $y^* = (0.01, 0.5, 0)^T$. The numerical results are summarized in table 3.4.

	$\psi_1(t)$	$\psi_2(t)$	$\psi_3(t)$	$\psi_4(t)$	$\psi_5(t)$
Inn	7	167	12	19	10
Out	8	8	8	8	8
Time	0.06	0.10	0.06	0.063	0.07

Table 3.4: Numerical results of problem 3.2.

Problem 3.3. The data (M, N, q) of the HLCP is given by

$$\begin{aligned} M &= (m_{ij}) = \begin{cases} -6 & \text{if } i = j \\ 2 & \text{if } |i-j| = 1 \quad \forall i,j = 1,...,n. \\ 0 & \text{otherwise} \end{cases}, \\ N &= (n_{ij}) = \begin{cases} -4 & \text{if } i = j \quad \forall i,j = 1,...,n. \\ 0 & \text{otherwise} \end{cases}, \\ q &= (4,2,...,2,4)^T . \end{aligned}$$

The exact solution is $x^* = (1, ..., 1)^T$ and $y = (0, ..., 0)^T$. The numerical results are summarized in the table bellow

	$n = 3$			$n = 100$			$n = 150$		
	Inn	**Out**	**Time**	**Inn**	**Out**	**Time**	**Inn**	**Out**	**Time**
$\psi_1(t)$	4	8	0.06	12	9	0.45	18	10	3.06
$\psi_2(t)$	179	8	0.11	70	9	9.07	1447	10	11.6
$\psi_3(t)$	15	8	0.06	52	9	0.19	77	10	0.91
$\psi_4(t)$	14	8	0.063	–	–	–	–	–	–
$\psi_5(t)$	8	8	0.068	23	9	0.63	28	10	2.30

Table 3.5: Numerical results of problem 3.3.

Problem 3.4. The $P_*(\kappa)$-HLCP is given by

$$M = \begin{pmatrix} 5 & 1 & 1 & 1 & 0 \\ 1 & 5 & 1 & 1 & 0 \\ 1 & 1 & 5 & 1 & 0 \\ 1 & 1 & 1 & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 5 \end{pmatrix}, N = nI, q = (21, 28, ..., 28, 21)^T,$$

The exact solution is $x^* = (0, ..., 0)^T$ and $y^* = (3, 4, ..., 3)^T$. We have the table:

	$n = 3$			$n = 50$			$n = 150$		
	Inn	**Out**	**Time(s)**	**Inn**	**Out**	**Time(s)**	**Inn**	**Out**	**Time(s)**
$\psi_1(t)$	3	8	0.07	10	9	0.12	19	10	1.73
$\psi_2(t)$	259	8	0.15	615	9	0.67	1301	10	13.96
$\psi_3(t)$	16	8	0.05	215	9	0.45	71	10	0.64
$\psi_4(t)$	18	8	0.05	–	–	–	–	–	–
$\psi_5(t)$	12	8	0.07	18	9	0.15	39	10	2.47

Table 3.6: Numerical results of problem 3.4.

The results obtained in Tables 3.3,3.4, 3.5 and 3.6 justify that the performance of the proposed kernel function seems quite promising. From the use of dynamic choice, it is clear that the iteration numbers presented in these tables can be reduced significantly compared to theoretical choice of the step-size.

3.4 Conclusion

In this chapter, we have proposed a primal-dual interior-point method with trigonometric kernel function to solve the $P_*(\kappa)$-HLCP. The corresponding algorithm processes the best known upper bound

complexity for large and small-update methods. We have provided computational experiments to confirm the efficiency of our approach. To accelerate the process of the algorithm, we have proposed the use of dynamic choice of the step-size. In this case, the algorithm produces better iteration numbers than the case of theoretical choice and significant decrease in the calculation time. For further research, this algorithm may be possible extended to other classes of problems.

Chapter 4

A Mehrotra type predictor-corrector interior-point method for $P_*(\kappa)$-HLCP

The denomination prediction-correction comes from the field of algorithms for solving differential equations, there is also a prediction step and a correction step. The principle of the prediction-correction method is to seek to achieve two objectives: to remain central and to approach the optimum using two type of different iterations. On the other hand the small-update methods seek to achieve these two objectives simultaneously. The predictor-corrector algorithm proposed by Mehrotra requires less computational effort than the other methods. In addition, the correction step uses some information from the prediction step. The idea of Mehrotra is to determine the parameter of centering adaptively and no longer according to a value fixed in advance as in path-following algorithms realizing the difficulty of improving proximity to the central path and reduction of the duality measure in a single step. The algorithm alternates between two different types of iterations, a prediction step to reduce the duality measure and a correction step to bring the iterate into the central path neighborhood.

4.1 Predictor-corrector algorithm for $P_*(\kappa)$-HLCP

Recalling the HLCP as follows

$$-Mx + Ny = q, \quad xy = 0, \quad (x, y) \geq 0, \tag{4.1}$$

where $M, N \in \mathbb{R}^{n\times n}, x, y, q \in \mathbb{R}^n$ and xy denotes the componentwise product of vectors x and y, with

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : Ny - Mx = q, x \geq 0, y \geq 0\}$$

$$\mathcal{F}^0 = \{(x, y) \in \mathcal{F} : x > 0, y > 0\}$$

Assumption 1 We assume that the HLCP satisfies the interior-point condition i.e, there exists a pair of vectors (x^0, y^0) such that $-Mx^0 + Ny^0 = q, (x^0, y^0) > 0$.

Assumption 2 The pair $\{M, N\}$ is a $P_*(\kappa)$-pair.

To solve the system (4.1) approximately, the following parameterized system is established

$$\begin{cases} Ny = Mx + q \\ xy = \mu e \\ x, y > 0, \end{cases} \tag{4.2}$$

where $\mu > 0$. It's known that under our assumptions [37], the system (4.1) has a unique solution for any $\mu > 0$. If $\mu \to 0$, $\{(x(\mu), y(\mu)) | \mu > 0\}$ approaches the solution of (4.1). Our algorithm will restrict the iterates to the negative infinity norm neighborhood of the central path:

$$\mathcal{N}_\infty^-(\gamma) = \left\{(x, y) \in \mathcal{F}^0 : x_i y_i \geq \gamma\mu, \forall i = 1, .., n\right\};$$

where $\gamma \in]0, 1[$ and $\mu = \frac{x^T y}{n}$. In the predictor step the algorithm computes the affine scaling search direction $(\Delta x^a, \Delta y^a)$, which is the solution of the following linear system

$$\begin{cases} -M\Delta x^a + N\Delta y^a = 0, \\ y\Delta x^a + x\Delta y^a = -xy, \end{cases} \tag{4.3}$$

then we compute the maximum predictor step length such that

$$\alpha_a = \max\{\alpha : (x + \alpha\Delta x^a, y + \alpha\Delta y^a) \geq 0, 0 \leq \alpha \leq 1\}.$$

However, this algorithm does not make a new iteration in the direction $(\Delta x^a, \Delta y^a)$. The algorithm uses some information from the prediction phase to calculate the corrector search direction $(\Delta x, \Delta y)$ by solving the system

$$\begin{cases} -M\Delta x + N\Delta y = 0 \\ y\Delta x + x\Delta y = \hat{\mu}e - xy - \alpha_a^2 \Delta x^a \Delta y^a, \end{cases} \tag{4.4}$$

where

$$\hat{\mu} = \frac{\gamma}{1 - \gamma}\mu.$$

The new iterate is defined by

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

with α is the corrector step length defined by

$$\alpha = \max\left\{\alpha \in [0,1] : (x(\alpha), y(\alpha)) \in \mathcal{N}_\infty^-(\gamma)\right\}.$$

The Mehrotra-type predictor-corrector algorithm for $P_*(\kappa)$-HLCP is as follows:

Algorithm

Input

A proximity parameter $\gamma \in [0,1]$;

A starting point $(x^0, y^0) \in \mathcal{N}_\infty^-(\gamma)$;

An accuracy parameter $\epsilon > 0$;

Set $k = 0$;

Begin

While $x^k y^k \geq \epsilon$ **do**

solve (4.3) to get $(\Delta x^a, \Delta y^a)$;

solve (4.4) to get $(\Delta x, \Delta y)$;

compute α the corrector step size;

set $\left(x^{k+1}, y^{k+1}\right) = \left(x^k, y^k\right) + \alpha\left(\Delta x^k, \Delta y^k\right)$;

set $k = k + 1$;

end while

end

4.2 Algorithmic analysis

In this section, we mainly characterize the polynomial complexity of Algorithm 1. We list some lemmas and theorems, which will play an important role in the complexity analysis.

Lemma 4.2.1. *Let* $(\Delta x^a, \Delta y^a)$ *be the solution of (4.3), then*

$$\begin{aligned} \Delta x_i^a \Delta y_i^a &\leq \frac{1}{4} x_i y_i, \forall i \in J^+ \\ \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a &\leq \frac{x^T y}{4}, \end{aligned}$$

where $J^+ = \{i \in J : \Delta x_i^a \Delta y_i^a \geq 0\}$, $J^- = \{i \in J : \Delta x_i^a \Delta y_i^a < 0\}$ *and* $J = \{1, 2, ..., n\}$.

Proof. From (4.3), for $i \in J^+$, we have

$$y_i \Delta x_i^a + x_i \Delta y_i^a = -x_i y_i,$$

dividing by $x_i y_i$

$$\frac{\Delta x_i^a}{x_i} + \frac{\Delta y_i^a}{y_i} = -1,$$

then using

$$0 \leq \left(\frac{\Delta x_i^a}{x_i} - \frac{\Delta y_i^a}{y_i} \right)^2 = \left(\frac{\Delta x_i^a}{x_i} \right)^2 + \left(\frac{\Delta y_i^a}{y_i} \right)^2 - 2 \frac{\Delta x_i^a \Delta y_i^a}{x_i y_i} = 1 - 4 \frac{\Delta x_i^a \Delta y_i^a}{x_i y_i},$$

we get

$$\Delta x_i^a \Delta y_i^a \leq \frac{x_i y_i}{4}.$$

Since $x_i > 0$ and $y_i > 0$, then, using previous inequality, we get

$$\sum_{i \in J^+} \Delta x_i^a \Delta y_i^a \leq \sum_{i \in J^+} \frac{x_i y_i}{4} \leq \frac{x^T y}{4}.$$

□

Lemma 4.2.2. *Let* (M, N) *be a pair of matrices defining a* $P_*(\kappa)$*-HLCP,* $(x, y) \in \mathcal{N}_\infty^-(\gamma)$ *and* $(\Delta x, \Delta y)$ *be the solution of (4.4) with* $\hat{\mu} > 0$*, then*

$$\|\Delta x \Delta y\| \leq \sqrt{\left(\frac{1}{4} + \kappa \right) \left(\frac{1}{2} + \kappa \right)} \left\| \hat{\mu} (xy)^{-\frac{1}{2}} - (xy)^{\frac{1}{2}} - \alpha_a^2 (xy)^{-\frac{1}{2}} \Delta x^a \Delta y^a \right\|^2.$$

Proof. First, we prove that

$$\sum_{i \in J^+} \Delta x_i \Delta y_i \leq \frac{1}{4} \left\| (xy)^{\frac{-1}{2}} \left(\hat{\mu} e - xy - \alpha_a^2 \Delta x^a \Delta y^a \right) \right\|^2, \tag{4.5}$$

by dividing the second equation in (4.4) with $(xy)^{1/2}$, we get

$$D\Delta x + D^{-1}\Delta y = (xy)^{\frac{-1}{2}}\left(\hat{\mu}e - xy - \alpha_a^2\Delta x^a\Delta y^a\right),$$

where $D = X^{\frac{-1}{2}}Y^{\frac{1}{2}}$ is a diagonal matrix. From previous relation, we get

$$0 < 4\Delta x_i\Delta y_i \leq \frac{\left(\hat{\mu} - x_iy_i - \alpha_a^2\Delta x_i^a\Delta y_i^a\right)^2}{x_iy_i}, \forall i \in J^+,$$

with the sum over $i \in J^+$, we get the desired inequality. On the other hand, since the problem is a $P_*(\kappa)$-HLCP, we have

$$(1+4\kappa)\sum_{i\in J^+}\Delta x_i\Delta y_i \geq -\sum_{i\in J^-}\Delta x_i\Delta y_i,$$

so that

$$\|\Delta x\Delta y\|_\infty \leq (1+4\kappa)\sum_{i\in J^+}\Delta x_i\Delta y_i$$

and

$$\|\Delta x\Delta y\|_1 \leq (2+4\kappa)\sum_{i\in J^+}\Delta x_i\Delta y_i,$$

using (4.5), we find

$$\|\Delta x\Delta y\|_\infty \leq \left(\frac{1}{4}+\kappa\right)\left\|(xy)^{\frac{-1}{2}}\left(\hat{\mu}e - xy - \alpha_a^2\Delta x^a\Delta y^a\right)\right\|^2$$

and

$$\|\Delta x\Delta y\|_1 \leq \left(\frac{1}{2}+\kappa\right)\left\|(xy)^{\frac{-1}{2}}\left(\hat{\mu}e - xy - \alpha_a^2\Delta x^a\Delta y^a\right)\right\|^2,$$

that implies

$$\|\Delta x\Delta y\|_2 \leq \sqrt{\left(\frac{1}{4}+\kappa\right)\left(\frac{1}{2}+\kappa\right)}\left\|(xy)^{\frac{-1}{2}}\left(\hat{\mu}e - xy - \alpha_a^2\Delta x^a\Delta y^a\right)\right\|^2.$$

□

Lemma 4.2.3. *If $(x,y) \in \mathcal{N}_\infty^-(\gamma)$, and $(\Delta x, \Delta y)$ is the solution of (4.4) with $\hat{\mu} = \frac{\gamma}{1-\gamma}\mu$, then*

$$\|\Delta x\Delta y\| \leq \sqrt{(1+4\kappa)(2+4\kappa)}An\mu,$$

such that

$$A = \frac{\gamma}{4(1-\gamma)^2} + \frac{29}{64} + \frac{\kappa}{4}.$$

Proof. From previous lemma we have

$$\|\Delta x \Delta y\| \leq \sqrt{\left(\frac{1}{4}+\kappa\right)\left(\frac{1}{2}+\kappa\right)}\left\|\hat{\mu}\left(xy\right)^{-\frac{1}{2}}-\left(xy\right)^{\frac{1}{2}}-\alpha_a^2\left(xy\right)^{-\frac{1}{2}}\Delta x^a \Delta y^a\right\|^2,$$

where

$$\begin{aligned}\left\|\hat{\mu}\left(xy\right)^{-\frac{1}{2}}-\left(xy\right)^{\frac{1}{2}}-\alpha_a^2\left(xy\right)^{-\frac{1}{2}}\Delta x^a \Delta y^a\right\|^2 &= \hat{\mu}^2\sum_{i=1}^{n}\frac{1}{x_i y_i}+\sum_{i=1}^{n}x_i y_i-2n\hat{\mu} \\ &\quad +\alpha_a^4\sum_{i=1}^{n}\frac{\left(\Delta x_i^a \Delta y_i^a\right)^2}{x_i y_i}-2\hat{\mu}\alpha_a^2\sum_{i=1}^{n}\frac{\Delta x_i^a \Delta y_i^a}{x_i y_i} \\ &\quad +2\alpha_a^2\sum_{i=1}^{n}\Delta x_i^a \Delta y_i^a,\end{aligned}$$

we have $x_i y_i \geq \gamma\mu$, then

$$\hat{\mu}^2\sum_{i=1}^{n}\frac{1}{x_i y_i}\leq\frac{\hat{\mu}^2 n}{\gamma\mu},$$

and from

$$\left(x+\alpha_a\Delta x^a\right)\left(y+\alpha_a\Delta y^a\right)\geq 0,$$

we have

$$-\alpha_a^2\Delta x^a \Delta y^a\leq xy,$$

and since the problem is a $P_*(\kappa)$ and from Lemma 4.2.1, we get

$$\begin{aligned}\alpha_a^4\sum_{i=1}^{n}\frac{\left(\Delta x_i^a \Delta y_i^a\right)^2}{x_i y_i} &= \alpha_a^4\sum_{i\in J^+}\frac{\left(\Delta x_i^a \Delta y_i^a\right)^2}{x_i y_i}+\alpha_a^4\sum_{i\in J^-}\frac{\left(\Delta x_i^a \Delta y_i^a\right)^2}{x_i y_i} \\ &\leq \sum_{i\in J^+}\frac{\left(\frac{x_i y_i}{4}\right)^2}{x_i y_i}+\sum_{i\in J^-}\left(-\alpha_a^2\frac{\Delta x_i^a \Delta y_i^a}{x_i y_i}\right)\left(-\alpha_a^2\Delta x_i^a \Delta y_i^a\right) \\ &\leq \frac{x^T y}{16}-\sum_{i\in J^-}\Delta x_i^a \Delta y_i^a \\ &\leq \frac{x^T y}{16}+\left(1+4\kappa\right)\sum_{i\in J^+}\Delta x_i^a \Delta y_i^a \\ &\leq \frac{x^T y}{16}+\left(1+4\kappa\right)\frac{x^T y}{4} \\ &= \left(\frac{5}{16}+\kappa\right)n\mu,\end{aligned}$$

and

$$-2\hat{\mu}\alpha_a^2\sum_{i=1}^{n}\frac{\Delta x_i^a \Delta y_i^a}{x_i y_i}\leq 2\hat{\mu}n,$$

and

$$\sum_{i=1}^{n} \Delta x_i^a \Delta y_i^a \leq \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a \leq \frac{x^T y}{4} = \frac{n\mu}{4}, \tag{4.6}$$

combining results, the lemma holds. □

Lemma 4.2.4. *Let* $(\Delta x^a, \Delta y^a)$ *be the solution of (4.3), then*

$$\Delta x^{aT} \Delta y^a \geq -\kappa x^T y.$$

Proof. Since the problem is $P_*(\kappa)$, then

$$\begin{aligned} -M\Delta x^a + N\Delta y^a &= 0 \Rightarrow (1+4\kappa) \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a + \sum_{i \in J^-} \Delta x_i^a \Delta y_i^a \geq 0 \\ &\Rightarrow \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a + \sum_{i \in J^-} \Delta x_i^a \Delta y_i^a + 4\kappa \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a \geq 0 \\ &\Rightarrow \Delta x^{aT} \Delta y^a \geq -4\kappa \sum_{i \in J^+} \Delta x_i^a \Delta y_i^a. \end{aligned}$$

From Lemma 4.2.1

$$\Delta x^{aT} \Delta y^a \geq -4\kappa \frac{x^T y}{4} = -\kappa x^T y.$$

□

Lemma 4.2.5. *Let* $(x, y) \in \mathcal{N}_\infty^- (\gamma), (\Delta x, \Delta y)$ *be the solution of (4.4) and* α *be the corrector step-size, then*

$$\alpha \leq \frac{\gamma}{(1+\gamma) n \sqrt{(1+4\kappa)(2+4\kappa) A}},$$

where

$$A = \frac{\gamma}{4(1-\gamma)^2} + \frac{29}{64} + \frac{\kappa}{4}.$$

Proof. We search α the corrector step-size such that $\alpha \in (0,1)$ and the new iterate $(x(\alpha), y(\alpha)) \in \mathcal{N}_\infty^- (\gamma)$, i.e $x_i(\alpha) y_i(\alpha) \geq \gamma\mu(\alpha), \forall i \in I = 1, ..., n$.

$$\begin{aligned} x_i(\alpha) y_i(\alpha) &= (x_i + \alpha\Delta x_i)(y_i + \alpha\Delta y_i) \\ &= x_i y_i + \alpha(x_i \Delta y_i + y_i \Delta x_i) + \alpha^2 \Delta x_i \Delta y_i \\ &= x_i y_i + \alpha\left(\hat{\mu} - x_i y_i - \alpha_a^2 \Delta x_i^a \Delta y_i^a\right) + \alpha^2 \Delta x_i \Delta y_i, \end{aligned}$$

we have $x_i y_i \geq \gamma\mu$, and using Lemma 4.2.1 and Lemma 4.2.3 we get

$$x_i(\alpha)\, y_i(\alpha) \geq \gamma\mu\left(1-\alpha-\alpha\frac{\alpha_a^2}{4}\right)+\alpha\frac{\gamma}{1-\gamma}\mu-n\sqrt{(1+4\kappa)(2+4\kappa)}A\mu\alpha^2,$$

and

$$\begin{aligned}\mu(\alpha) &= \frac{(x+\alpha\Delta x)^T(y+\alpha\Delta y)}{n}\\ &= \mu+\alpha\hat{\mu}-\alpha\mu-\alpha\alpha_a^2\frac{\Delta x^{a^T}\Delta y^a}{n}+\alpha^2\frac{\Delta x^T\Delta y}{n},\end{aligned}$$

using Lemma 4.2.3 and Lemma 4.2.4, we get

$$\mu(\alpha) \leq \mu+\alpha\frac{\gamma}{1-\gamma}\mu-\alpha\mu+\alpha\alpha_a^2\kappa\mu+n\sqrt{(1+4\kappa)(2+4\kappa)}A\mu\alpha^2,$$

to find α, we put

$$\gamma\mu\left(1-\alpha-\alpha\frac{\alpha_a^2}{4}\right)+\alpha\frac{\gamma}{1-\gamma}\mu-n\sqrt{(1+4\kappa)(2+4\kappa)}A\mu\alpha^2 \geq$$

$$\gamma\left(\mu+\alpha\frac{\gamma}{1-\gamma}\mu-\alpha\mu+\alpha\alpha_a^2\kappa\mu+n\sqrt{(1+4\kappa)(2+4\kappa)}A\mu\alpha^2\right)$$

and

$$(1+\gamma)\, n\sqrt{(1+4\kappa)(2+4\kappa)}A\alpha \leq \gamma\left(1-\frac{\alpha_a^2}{4}-\alpha_a^2\kappa\right) \leq \gamma$$

then

$$\alpha \leq \frac{\gamma}{(1+\gamma)\, n\sqrt{(1+4\kappa)(2+4\kappa)}A} = \bar{\alpha}$$

we define $\bar{\alpha}$ as a default corrector step-size. □

Theorem 4.2.1. *The algorithm will stop in* $O\left(\frac{nA\sqrt{(1+4\kappa)(2+4\kappa)}}{\Gamma}\log\frac{1}{\epsilon}\right)$ *iteration complexity, where*

$$A = \frac{\gamma}{4(1-\gamma)^2}+\frac{29}{64}+\frac{\kappa}{4},$$

$$\Gamma = \left(\frac{\gamma}{1+\gamma}\right)\left(1-\kappa-\frac{2\gamma}{1-\gamma^2}\right).$$

Proof.

$$\mu(\alpha) = \frac{x^Ty}{n}+\frac{\alpha}{n}\left(\hat{\mu}n-x^Ty-\alpha_a^2\Delta x^{a^T}\Delta y^a\right)+\frac{\Delta x^T\Delta y}{n},$$

using Lemma 4.2.3, Lemma 4.2.4 and Lemma 4.2.5, we get

$$\begin{aligned}
\mu(\alpha) &\leq \mu + \alpha\left(\frac{\gamma}{1-\gamma}\mu - \mu + \alpha_a\kappa\mu\right) + \alpha^2 nA\sqrt{(1+4\kappa)(2+4\kappa)}\mu \\
&\leq \mu\left(1-\alpha\left(1-\kappa-\frac{\gamma}{1-\gamma}-\alpha n\sqrt{(1+4\kappa)(2+4\kappa)}A\right)\right) \\
&\leq \mu\left(1-\alpha\left(1-\kappa-\frac{2\gamma}{1-\gamma^2}\right)\right) \\
&\leq \mu\left(1-\left(\frac{1}{n\sqrt{(1+4\kappa)(2+4\kappa)}A}\left(\frac{\gamma}{1+\gamma}\right)\left(1-\kappa-\frac{2\gamma}{1-\gamma^2}\right)\right)\right) \\
&= \mu\left(1-\frac{\Gamma}{n\sqrt{(1+4\kappa)(2+4\kappa)}A}\right),
\end{aligned}$$

such that

$$\Gamma = \left(\frac{\gamma}{1+\gamma}\right)\left(1-\kappa-\frac{2\gamma}{1-\gamma^2}\right),$$

we must get

$$\mu(\alpha) \leq \epsilon\mu^0,$$

then

$$\left(1-\frac{\Gamma}{nA\sqrt{(1+4\kappa)(2+4\kappa)}}\right)^k \mu^0 \leq \epsilon\mu^0,$$

this completes the proof. □

4.3 Numerical results

In this section, we present some numerical results where the algorithm is coded in MATLAB R2009b and our experiments were performed on PC with the following parameters: $\gamma = 0.01$, $\epsilon = 10^{-5}$ the tolerance and the initial point is $x^0 = y^0 = e$. In what follows, we use the dynamic step-size presented in [50]

$$\alpha_d = \begin{cases} 2\alpha, \|\Delta x\| \geq n; \\ 5\alpha, 1 < \|\Delta x\| < n; \\ 10\alpha, \|\Delta x\| < 1 \end{cases}$$

and the practical step-size presented in [36]

$$\alpha_x = \begin{cases} \min\limits_i\left(-\frac{\Delta x_i}{x_i}\right) & \text{if } (\Delta x_i) < 0, \forall i = 1, ..., n \\ 1 & \text{otherwise} \end{cases}$$

$$\alpha_y = \begin{cases} \min\limits_i \left(-\frac{\Delta y_i}{y_i}\right) & \text{if } (\Delta y_i) < 0, \forall i = 1, ..., n \\ 1 & \text{otherwise} \end{cases}$$

$$\alpha_p = \min(\alpha_x, \alpha_y).$$

We consider the following problems:

Problem 4.1. Consider an HLCP with the following setting:

$$M = \begin{pmatrix} 1.5 & 10.5 & 1.05 & 1.05 & 2 & 0 & 0 \\ 2 & 1.5 & 6 & 6 & 1.5 & 1.5 & 2 \\ 1.5 & 3.5 & 55 & 9.5 & 100 & 1500 & -5 \\ 5 & 1.5 & 3.5 & 1 & 0 & 3 & 40.5 \\ 3 & 3 & 8.25 & 2.5 & 2.25 & 0 & 2.5 \\ 1.5 & 5 & 5 & 0 & 0 & 1.05 & 0 \\ 1.5 & 1.05 & 1 & 1.05 & 1 & 2 & 1000 \end{pmatrix},$$

$$N = \begin{pmatrix} 0.5 & 9.5 & 0.95 & 0.95 & 2 & 0 & 0 \\ 2 & 0.5 & 6 & 6 & 0.5 & 0.5 & 2 \\ 0.5 & 2.5 & 4.5 & 8.5 & 100 & 1500 & -5 \\ 5 & 0.5 & 2.5 & 1 & 0 & 3 & 39.5 \\ 3 & 3 & 7.75 & 1.5 & 1.75 & 0 & 1.5 \\ 0.5 & 5 & 5 & 0 & 0 & 0.95 & 0 \\ 0.5 & 0.95 & 1 & 0.95 & 1 & 2 & 1000 \end{pmatrix}$$

$$q = (-16.4, 23, 3206, 79, 13, -1.1, 2004.8)^T.$$

The exact solution is $x^* = (2, 2, 0, 0, 0, 0, 0)^T$ and $y^* = (0, 0, 2, 2, 2, 2, 2)^T$. The numerical results are summarized in the table bellow:

	Iterations	**Time(s)**
Theoretical step-size	1796	0.61
Dynamic step-size	32	0.069
Practical step size	5	0.02

Table 4.1: Numerical results of problem 4.1.

Problem 4.2. For $c \geq 0$, the $P_*(\kappa)$-HLCP is given by

$$M = \begin{pmatrix} 0 & 1+4\kappa & 0 \\ -1 & 0 & 0 \\ 0 & 0 & c \end{pmatrix}, N = I \text{ and } q = (0.01, 0.501, -0.49)^T.$$

The problem is a $P_*(\kappa)$-HLCP for all $\kappa \geq 0$. For example if $\kappa = 1$ and $c = 1$, then the HLCP (a non monotone HLCP) is given by

$$M = \begin{pmatrix} 0 & 1+4 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, N = I \text{ and } q = (0.01, 0.501, -0.49)^T.$$

The exact solution $x^* = (0, 0, 0.49)^T$ and $y^* = (0.01, 0.5, 0)^T$.

	Iterations	**Time(s)**
Theoretical step-size	661	0.46
Dynamic step-size	31	0.06
Practical step size	7	0.005

Table 4.2: Numerical results of problem 4.2.

Problem 4.3 The HLCP is given by

$$M = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix}, N = I \text{ and } q = (-1, ..., -1)^T.$$

The exact solution is $x^* = (0.3660, 0.4639, 0.4897, 0.4948, 0.4897, 0.4639, 0.3660)^T$ and $y^* = (0, ..., 0)^T$. The numerical results are in the following table:

	Iterations	Time (s)
Theoretical step-size	7180	2.81
Dynamic step-size	20	0.027
Practical step-size	4	0.007

Table 4.3: Numerical results of problem 4.3.

Problem 4.4.

$$M = \begin{pmatrix} -4 & 8 & -8 \\ -3 & 1 & -1 \\ 4 & -3 & 2 \end{pmatrix}, N = \begin{pmatrix} 1 & -2 & 3 \\ 2 & 1 & 1 \\ -2 & 0 & -1 \end{pmatrix} \text{ and } q = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}.$$

The exact solution $x^* = (0, 0.5, 0)^T$ and $y^* = (0.5, 0, 0.5)^T$.

	Iterations	Time (s)
Theoretical step-size	3180	0.69
Dynamic step-size	53	0.029
Practical step-size	7	0.005

Table 4.4: Numerical results of problem 4.4.

Problem 4.5. [63] The HLCP is given by

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & \cdots & 6 \\ 2 & 6 & 9 & \cdots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \cdots & 4n-3 \end{pmatrix}, N = I, \text{ and } q = (-1, -1, \ldots, -1, -1)^T.$$

The exact solution is $x^* = (1, 0, ..., 0)^T$ and $y^* = (0, 1, ..., 1)^T$.

The numerical results are in the following table:

Theoretical step-size			dynamic step-size			practical step-size		
n	**Iterations**	**Time(s)**	n	**Iterations**	**Time(s)**	n	**Iterations**	**Time(s)**
3	171	0.16	3	9	0.05	3	2	0.01
10	619	0.92	10	12	0.09	10	2	0.014
100	6974	37.13	100	61	0.059	100	3	0.05
1000	67440	359.05	1000	130	30.25	1000	3	11.54
3000	-	-	3000	-	-	3000	3	284.23

Table 4.5: Numerical results of problem 4.5.

Next, we compare our algorithm which we denoted by **Algo3** using the practical step-size with Mehrotra-type predictor-corrector algorithm proposed by Zhou et al. [63] in 2019 denoted by **Algo4**. From [63], we set $\epsilon = 10^{-8}$ and $\gamma = 0.01$.

n	Algorithms	Iterations
10	**Algo3**	2
	Algo4	10
20	**Algo3**	2
	Algo4	11
30	**Algo3**	2
	Algo4	12
40	**Algo3**	2
	Algo4	13
50	**Algo3**	2
	Algo4	13
100	**Algo3**	3
	Algo4	15
150	**Algo3**	3
	Algo4	15
200	**Algo3**	3
	Algo4	16

Table 4.6: Numerical results of problem 4.3.

The results in these tables show that the algorithm based on the practical step-size is efficient. The number of iterations and the calculation time have significantly reduced. While the algorithm based on theoretical step-size may leads to small step size so more iterations. In addition, the numerical results in table (4.6) show that our algorithm performs well in comparison with the algorithm in [63].

4.4 Conclusion

In this chapter, we have proposed a new feasible version of Mehrotra's predictor-corrector algorithm. Whose variants have been widely used in several interior-point methods based optimization packages. We have demonstrate the polynomial complexity of the algorithm. Also, we have presented some numerical tests to examine its practical efficiency.

General conclusion

In this thesis, we consider to solve horizontal linear complementarity problems (HLCPs) with $P_*(\kappa)$-matrix. LCP is not an optimization problem. However, KKT conditions of several important optimization problems, such as linear and quadratic programming problems (LP and QP), can be formulated as LCP. In addition, many problems from engineering, transportation, finance, etc. can be written as LCP. Hence, it is of significant interest to consider LCP and find efficient methods to solve it. The class of methods that have shown to be very efficient and with polynomial worst case complexity is called interior-point methods (IPMs). In this thesis, we consider primal-dual cental-path IPM applied to $P_*(\kappa)$-horizontal linear complementarity problem (HLCP).

HLCP arises in different domains and also has many applications in mathematics. So, first, we have presented a theoretical and practical study of the transformation of an absolute value equation to an HLCP by introducing an infeasible primal-dual central path method.

Then, we have proposed a primal-dual interior-point method based on a new class of parametric kernel functions, this approach have the advantage of starting with any strictly feasible point not necessary close to the central path. Based on this function, a class of large and small-update primal–dual interior-point algorithms for HLCP is proposed. The complexity analysis shows that the iteration bounds for the small and large-update primal–dual IPMs based on this function coincide to the so far best known iteration complexities, i.e. $O\left((1+2\kappa)\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$ for large-update and $O\left((1+2\kappa)\sqrt{n}\log\frac{n}{\epsilon}\right)$ for small-update methods. Moreover, we have presented some comparative numerical results to show the practical behaviour of the new proposed kernel functions. The numerical experiments demonstrate that the algorithm based on the new kernel function reduces the number of iterations and CPU time.

Finally, a new variant of Mehrotra-type predictor-corrector algorithm was proposed. This type of algorithm is presented in all available optimization software. Indeed, this algorithm allows to improve the convergence and to present one of the best results of the interior-point methods complexity in optimization. We have provided some numerical results to illustrate the advantage of our new algorithm and we have compared our algorithm with Mehrotra-type predictor-corrector algorithm presented by Zhou et al. [63]. This comparison reveals that for all of the selected problems our proposed algorithm

works practically better, as well.

Several questions remain open for future work:

- The generalization of the presented algorithms to more general problems such as the sufficient LCP and the Cartesian $P_*(\kappa)$-LCPs over symmetric cones.
- Extending the computational study of these algorithms to some wider classes of LCPs (coming from bimatrix games, etc).
- Another important question would be how to find a general class of kernel functions which gives interior-point algorithms with the best known complexity results for solving $P_*(\kappa)$-HLCP.

This work is supported by the following publications:

- N. Hazzam and Z. Kebbiche, A primal-dual interior point method for $P_*(\kappa)$-HLCP based on a class of parametric kernel functions, *Numerical Algebra Control and Optimization*, (2019).

- M. Achache and N. Hazzam, Solving general absolute value equations using complementarity and interior-point methods, *Journal of Nonlinear Functional Analysis*, (2018).

Bibliography

[1] M. Achache, "Complexity analysis of a weighted-full-Newton step interior-point algorithm for $P_*(\kappa)$-LCP", *RAIRO-Oper. Res.*, 50 (2016) 131–143.

[2] M. Achache, N. Hazzam, "Solving general absolute value equations using complementarity and interior-point methods", *Journal of Nonlinear Functional Analysis*, (2018).

[3] M. Achache, N. Tabchouche, "Complexity analysis and numerical implementation of large-update interior-point methods for SDLCP based on a new parametric barrier kernel function", *Optimization. A Journal of Mathematical Programming and Operations Research*, (2018).

[4] A. Arif, "Un algorithme prédicteur-correcteur pour le problème complémentaire linéaire", *Mémoire de Master* 2*, Université Ferhat Abbas, Sétif1, Algérie*, 2020.

[5] S. Asadi and H. Mansouri, "Polynomial interior-point algorithm for $P_*(\kappa)$-horizontal linear complementarity problems", *Numer. Algorithms*, **63**(2) (2013) 385–398.

[6] S. Asadi and H. Mansouri, "A Mehrotra type predictor-corrector interior-point algorithm for linear programming", *Numerical Algebra Control and Optimization*, (2019) 147–156.

[7] S. Asadi, H. Mansouri and M. Zangiabadi, "A class of path-following interior-point methods for $P_*(\kappa)$-horizontal linear complementarity problems", *J. Oper Res Soc China*, **3**(1) (2015) 17–30.

[8] S. Asadi, M. Zangiabadi and H. Mansouri, "An infeasible interior-point algorithm with full-Newton steps for $P_*(\kappa)$-horizontal linear complementarity problems based on a kernel function", *J. Appl. Math. Comput.* **50**(1) (2016) 15–37.

[9] R. Almeida and A. Teixeira, "On the convergence of a predictor-corrector variant algorithm", *TOP*, 23(2) (2014) 401–418.

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

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

[12] YQ. Bai, C. Roos, "A primal-dual interior-point method based on a new kernel function with linear growth rate", *Proceedings of the 9th Australian Optimization Day*, (2002).

[13] J.F. Bonnans and C.C. Gonzaga, "Convergence of interior-point algorithms for the monotone linear complementarity problem", *Mathematics of Operations Research*, 21 (1996)1-25.

[14] M. Bouafia, D. Benterki, A. Yassine, "An Efficient Primal–Dual Interior Point Method for Linear Programming Problems Based on a New Kernel Function with a Trigonometric Barrier Term", *J. Optim. Theory Appl.*, (2016).

[15] S. P. Dirkse and M. C. Ferris, "MCPLIB: A collection of non linear mixed complementarity problems", *Optlmizaiion Methods and Software*, 5(4) (1995) 319–345.

[16] Ch. Chennouf, "The effect of the step-size on the numerical behavior of a primal-dual interior-point algorithm applied to $P_*(\kappa)$-linear complementarity problem", *Bulletin of the Transilvania University of Brasov*, 13(62) (2020) 331–342.

[17] GM. Cho, MK. Kim, "A new large-update interior point algorithm for $P_*(\kappa)$-LCPs based on kernel functions", *Appl. Math. Comput.*, 182 (2006) 1169–1183.

[18] GM. Cho, "A new large-update interior point algorithm for $P_*(\kappa)$ linear complementarity problems", *J. Comput. Appl. Math.*, 216 (2008) 256–278.

[19] G.M. Cho, "Large-update interior point algorithm for P_*-linear complementarity problem", *J. Inequalities Appl.*, 363 (2014) 1–12.

[20] RW. Cottle, JS. Pang, RE. Stone, "The Linear Complementarity Problem", *Academic Press, San Diego*, (1992).

[21] J. Czyzyk, S. Mehrotra and M. Wagner, " S. Wright, An interior-point code for linear programming", *Optim. Methods Software*, 11(1-–4) (1999) 397-–430.

[22] L. Derbal and Z. Kebbiche, " An efficient parametrized logarithmic kernel function for semidefinite optimization", *Acta Mathematicae Applicatae Sinica, English Series*, 36(3) (2020) 753–770.

[23] M. EL Ghami and T. Steihaug, "Kernel-function based primal-dual algorithms for $P_*(\kappa)$ linear comlementarity problem", *RAIRO-Oper. Res.*, 44 (2010) 185–205.

[24] M. El Ghami, GQ. Wang, "Interior-point methods for $P_*(\kappa)$ linear comlementarity problem based on generalized trigonometric barrier function", *International Journal of Applied Mathematics*, (2017) 11–33.

[25] M. C. Ferris and J. S. Pang, "Engineering and economic applications of complementarity problems", *SIAM Review* , 39(4) (1997) 667–713.

[26] C. Gonzaga, " Path following methods for linear programming", *SIAM Review*, 34 (1992) 167–227.

[27] S. Fathi-Hafshejani and A. Fakharzadeh Jahromi, "An interior point method for $P_*(\kappa)$-horizontal linear complementarity problem based on a new proximity function", *J. Appl. Math. Comput.*, 62 (2020) 281–300.

[28] S.Hafshejania , H.Mansouri, M. Peyghami, "An interior-point algorithm for $P_*(\kappa)$-linear complementarity problem based on a new trigonometric kernel functionc", *Journal of Mathematical Modeling*, 5(2) (2017) 171–197.

[29] N. Hazzam and Z. Kebbiche, "A primal-dual interior point method for $P_*(\kappa)$-HLCP based on a class of parametric kernel functions", *Numerical Algebra, Control and Optimization*, (2019).

[30] R.A. Horn and C.R. Johnson, "Matrix Analysis", *Cambridge University Press*. (1990).

[31] T. Illés and M. Nagy, "A Mizuno–Todd–Ye type predictor–corrector algorithm for sufficient linear complementarity problems", *European Journal of Operational Research*, 181 (2007) 1097—1111.

[32] F. Jarre and M. Wechs, " Extending Mehrotra's corrector for linear programs", *Adv. Model. Optim.* 1(2) (1999) 38-–60.

[33] P. Ji, M. Zhang, X. Li, "A Primal-dual Large-update Interior-point Algorithm for $P_*(\kappa)$-LCP Based on a New Class of Kernel Functions", *Acta Mathematicae Applicatae Sinica, English Series*, Vol. 34, No.1 (2018) 119–134.

[34] N.K. Karmarkar, "A new polynomial-time algorithm for linear programming", *Combinatorica*, 4 (1984) 373—395.

[35] Z.Kebbiche, "Etude et extension d'algorithmes de points intérieurs pour la programmation nonlinéaire", *Thèse de doctorat, Université Ferhat Abbas de Sétif*, (2007).

[36] A. Keraghel, " Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar", "Thèse de Doctorat, Université Joseph Fourier-Grenoble I, France," (1989).

[37] M. Kojima, N. Megiddo, T.Noma, A. Yoshise, "A unified approach to interior point algorithms for linear complementarity problems: A summary", *Operations Research Letters*, 10(5) (1991) 247-254.

[38] M. Kojima, S. Mizuno, A. Yoshise, " A primal-dual interior point algorithm for linear programming", *Interior Point and Related Methods[M]. New York: Springer-Verlag*, (1989) 29–47.

[39] Y. Lee, Y. Cho, G. Cho, "Kernel function based interior-point methods for horizontal linear complementarity problems", *Journal of Inequalities and Applications*, 215 (2013).

[40] G. Lesaja, C. Roos, "Unified Analysis of Kernel-Based Interior-Point Methods for $P_*(\kappa)$–LCP", *SIAM Journal on Optimization*, Vol. 21 (2011) 2994—3013.

[41] H. Liu, X. Liu and C. Liu, "Mehrotra-type predictor—corrector algorithms for sufficient linear complementarity problem, *Appl. Numer. Math.* 62(12) (2012) 1685—1700.

[42] X. Liu, H. Liu, W. Wang, "Polynomial convergence of Mehrotra-type predictor–corrector algorithm for the Cartesian $P_*(\kappa)$-LCP over symmetric cones", *Optimization*, 64(4) (2015) 815—837.

[43] N. Megiddo "Interior Point and Related Methods", *Progress in Mathematical Programming New York: Springer-Verlag*, (1989) 131–158.

[44] S. Mehrotra, "On the Implementation of a Primal-Dual Interior-Point Method", *SIAM J. Optimization*, 2(4) (1992) 575–601.

[45] J. Miao, "A quadratically convergent $O\left((\kappa+1)\sqrt{n}L\right)$-iteration algorithm for the $P_*(\kappa)$-matrix linear complementarity problem", *Mathematical programming*, 69(1–3) (1995) 355–368.

[46] S. Mizuno, M. J. Todd and Y. Ye, "On Adaptive-Step Primal-Dual Interior-Point Algorithms for Linear Programming", *Mathematics of Operations Research*, 18(4) (1993) 964–981.

[47] J. Peng, C. Roos, T.Terlaky, "A new and efficient large-update interior point method for linear optimization", *J. Comput. Technol.*, 6 (2001) 61–80.

[48] J. Peng, C. Roos, T.Terlaky, "Self-regular functions and new search directions for linear and semidefinite optimization", *Math. Program., Ser.* 93 (2002) 129–171.

[49] F. A. Potra, "The Mizuno-Todd-Ye algorithm in a larger neighbourhood of the central path", *European Journal of Operational Research*, 143(2) (2002) 257–267.

[50] Z.G.Qian, Y.Q.Bay, "Primal-dual interior point algorithm with dynamic step-size based on kernel function for linear programming", *Journal of Shnghai University (English Edition)*, 9(5) (2005) 391-396.

[51] M. Reza Peyghami, S. Fathi Hafshejani, L. Shirvani, "Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function", *Journal of Computational and Applied Mathematics*, (2014).

[52] C. Roos, T. Terlaky and J. Ph. Vial, "Theory and Algorithms for Linear optimization. An Interior Point Approach", *John Wiley and Sons, Chichester*, (1997).

[53] H. Roumili, and N. Boudjellal, "An infeasible interior point methods for convex quadratic problems", *J. Numer. Anal. Approx. Theory*, 47(2) (2019) 177-186.

[54] M. Salahi and N. M. Amiri, "Polynomial time second order Mehrotra-type predictor—corrector algorithms", *Applied Mathematics and Computation* 183 (2006) 646—658.

[55] M. Salahi, J. Peng and T. Terlaky, "On Mehrotra-type predictor-corrector algorithms", *SIAM J. Optim.* 18(4) (2007) 1377-–1397.

[56] E.M. Simantiraki and D.F. Shanno, " An infeasible-interior point method for linear complementarity problems ", *SIAM J. Optim.* 7(3) (1997) 620-640.

[57] GQ. Wang, YQ. Bai, "Polynomial interior-point algorithm for $P_*(\kappa)$ horizontal linear complementarity problem", *J. Comput. Appl. Math.* 233 (2009) 248–263.

[58] S.J. Wright, "Primal-dual interior point methods", *Copyright by SIAM*. Philadelphia (1997).

[59] X. Yang, H. Liu and X. Dong, "Polynomial convergence of Mehrotra-type predictor-corrector infeasible-IPM for symmetric optimization based on the commutative class directions", *Appl. Math. Comput.*, 230(2) (2014) 616-–628.

[60] L. Yong, S. Zhang, F. Deng and W. Xiong, "Feasible Interior Point Method for Absolute Value Equation", *Fourth International Conference on Information and Computing*, (2011).

[61] Y. Zhang, "On the convergence of a class of infeasible interior-point for the horizontal linear complementarity problem", *SIAM Journal on Optimization*, 4 (1) (1994) 208–227.

[62] Y. Zhang, "Solving large-scale linear programs by interior-point methods under the Matlab environment", *Optim. Methods Software*, 10(1) (1998) 1—31.

[63] Y. Zhou, M. Zhang and Z. Huang, "On complexity of a new Mehrotra-type interior-point algorithm for $P_*(\kappa)$ linear complementarity problem", *Journal of Inequalities and Applications*, (2019) 1–13.

ملخص

نهتم في هذه الأطروحة بالتحليل والدراسة العددية لحل مشكلة التتام الخطي الأفقي باستعمال طريقة النقطة الداخلية. نقدم في الجزء الأول دراسة نظرية وعملية لتحويل معادلة القيمة المطلقة إلى مشكلة التتام الخطي الأفقي باستعمال طريقة المسار المركزي. في الجزء الثاني نقدم خوارزمية النقط الداخلية المعتمدة على دالة نواة جديدة ثم نوضح فعاليتها ببعض النتائج العددية. في الجزء الثالث يتم اقتراح بديل جديد لخوارزمية متنبئ-مصحح حيث ندرس تكلفتها الحدودية ثم نختبر الكفاءة العملية لهذه الخوارزمية عن طريق إجراء بعض الاختبارات الحسابية.

الكلمات المفتاحية: مشكلة التتام الخطي الأفقي ، طريقة النقط الداخلية، دالة نواة، خوارزمية متنبئ-مصحح، خوارزمية ميهروترا، تكلفة حدودية.

Résumé

Dans cette thèse, on s'intéresse à l'analyse et l'étude numérique des méthodes de points intérieurs pour résoudre le problème de complémentarité linéaire horizontal (PCLH). Dans la première partie, nous présentons une étude théorique et pratique de la transformation d'une équation en valeurs absolues à un PCLH en introduisant une méthode de trajectoire central primale-duale non réalisable. Dans la deuxième partie, on présente une méthode de points intérieurs primale-duale basée sur une nouvelle classe de fonctions noyaux. La complexité algorithmique prouvée pour cet algorithme est la meilleure complexité connue jusqu'à présent. Ensuite, on illustre la performance des fonctions noyaux proposées par quelques résultats numériques comparatifs. Dans la troisième partie, une nouvelle variante de l'algorithme de Mehrotra de type prédicteur-correcteur est proposée où sa complexité est prouvée polynomiale. Finalement, on teste l'efficacité pratique de l'algorithme en exécutant quelques tests numériques.

Mots clés : Problème de complémentarité linéaire horizontal, méthodes de points intérieurs, fonction noyau, algorithme prédicteur-correcteur, algorithme de Mehrotra, complexité algorithmique.

Abstract

In this thesis we are interesting in the analysis and numerical study of interior-point methods for solving horizontal linear complementarity problem (HLCP). In the first part, we presente a theoretical and practical study of the transformation of an absolute value equations to an HLCP by introducing an infeasible primal-dual central path method. In the second part, primal-dual interior-point methods based on a new class of kernel functions are presented. We show that the corresponding algorithm has the best known iteration bound for large-update methods. Then, we illustrated the performance of the proposed kernel functions by some comparative numerical results. In the third part, a new variant of Mehrotra type predictor-corrector algorithm is proposed where we prove that its complexity is polynomial. Finaly, We test the practical efficiency of the algorithm by running some computational tests.

Keywords: Horizontal linear complementarity problem, interior-point methods, kernel function, predictor-corrector algorithm, Mehrotra's algorithm, complexity bound.