Filomat 38:24 (2024), 8461–8479
https://doi.org/10.2298/FIL2424461D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

An interior-point algorithm for $P_*(\kappa)$-LCPs based on a new kernel function with a double barrier term

Louiza Derbal[a], Zakia Kebbiche[a]

[a] *Laboratory of Fundamental and Numerical Mathematics (LFNM), Department of Mathematics, Ferhat Abbas University, Setif 1. Setif 19000, Algeria*

Abstract. In this paper, new search directions and proximity measures are proposed for $P_*(\kappa)$-linear complementarity problem. The new method is based on a new class of kernel function which differs from the existing kernel functions in which it has a double barrier term. We prove that the interior-point algorithm has the same complexity bound as the best known interior-point algorithms for solving these types of problems. The corresponding algorithm has $O((1+2\kappa)\,q\,\sqrt{n}(\log\,\sqrt{n})^{\frac{q+1}{q}}\,\log\frac{n}{\epsilon})$ iteration complexity for large-update methods and we match the best known iteration bounds with special choice of the parameter q for $P_*(\kappa)$-linear complementarity problem that is $O((1+2\kappa)\,\sqrt{n}\log n\log\frac{n}{\epsilon})$.We illustrate the performance of the proposed kernel function by some comparative numerical results that are derived by applying our algorithm to an other kernel function.

1. Introduction

In this paper, we consider the following linear complementarity problem (*LCP*):

$$\begin{cases} s = Mx + q, \\ xs = 0, \\ x \geq 0, s \geq 0. \end{cases} \tag{1.1}$$

where $M \in \mathbb{R}^{n\times n}$ is a $P_*(\kappa)$-matrix and $q, x, s \in \mathbb{R}^n$, and xs denotes the componentwise product (Hadamard product) of vector x and s.

Note that (1.1) is a feasibility problem, and not an optimization problem. However,it is well-known that is can easily be writen as an optimization problem:

$$\begin{cases} \min x^T s \\ -Mx + s = q, \\ x \geq 0, s \geq 0. \end{cases} \tag{1.2}$$

2020 *Mathematics Subject Classification*. Primary 65K05; Secondary 90C33.
Keywords. Interior-point, kernel function, primal-dual method, large update, small update, linear complementarity problem.
Received: 23 July 2023; Revised: 12 April 2024; Accepted: 08 June 2024
Communicated by Predrag Stanimirović
Research supported by the General Directorate of Scientific Research and Technological Development (DGRSDT) of Algeria.
Email addresses: louiza.derbal@univ-setif.dz (Louiza Derbal), zakia.kebbiche@univ-setif.dz (Zakia Kebbiche)

LCPs arises in many areas such as variational inequalities, economic equilibrium problems and bimatrix games. It is known that this problem trivially includes the two important domains in optimization: the linear programming (LP) and the convex quadratic programming (CQP) in their usual formulations, then this problem becames the subject of many research interest. The aim of researchers is to extend all results obtained in LP and QP to a more general class of problems known as monotone LCPs, i.e., $P_*(0)$-LCP, and vice versa [26]. Feasible path-following algorithms are the most attractive interior point methods (IPMs) for solving a large wide of optimization problems. These algorithms achieved beautiful results such as polynomial complexity and numerical efficiency [21], [26]. They start with a strictly feasible centered starting point and maintain feasibility during the solution process. However, in practice these algorithms do not have always a strictly feasible centered starting point.The choice of the barrier function plays an important role not only in the analysis but also for the performance of the algorithm. Peng et al. [18] introduced primal-dual IPMs for linear optimization (LO) based on a new class of barrier functions that is defined by 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, $O(\sqrt{n}\log n\log\frac{n}{\epsilon})$. Bai et al. [2] proposed a new class of kernel functions which is not necessarily self-regular. The class is called eligible kernel function which greatly simplifies the analysis of primal-dual IPMs for LO. For some of them are matched the best iteration bounds for large update IPMs. Later, Bai et al. [3], Wang and Bai [22] generalized primal-dual IPMs for LO to sufficient LCP based on some specific eligible kernel functions. Cho et al. [4], [5], [6] extended the complexity analysis for LO problems to $P_*(\kappa)$-LCPs. Amini et al. [1] and Lee et al. [15] generalized versions of the kernel functions in [2] and improved the complexity results for large-update methods for $P_*(\kappa)$-LCPs. Recently, Lesaja et al. [16], [17] provided a unified approach and comprehensive analysis of IPMs for $P_*(\kappa)$-LCP and Cartesian $P_*(\kappa)$)-LCP over symmetric cones (SCLCP), respectively. Wang and Bai [23] presented a new class of polynomial IPMs for the Cartesian P-matrix SCLCP based on a parametric kernel function, which matched the currently best known iteration bounds for large- and small-update methods. Very recently, S. Fathi-Hafshejani et al. [11], [12], M. El Ghami et al. [10] presented a large-update IPM for $P_*(\kappa)$-LCP based on new trigonometric kernel function and showed that the iteration complexity is $O\left((1+2\kappa)\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$. For more studies with $P_*(\kappa)$-LCP you can refer to [7], [15], [19], [24], [25], [27].

In this paper, we focus on the complexity analysis of large-update IPMs for $P_*(\kappa)$-LCP based on a new kernel function with a double barrier term (combination of the classic kernel function and a barrier term), defined by:

$$\psi(t)=\frac{t^2-1-\log t}{2}+\frac{e^{\frac{1}{t^q}-1}-1}{2q}\qquad\text{for}\quad t>0,q\geq 1. \tag{1.3}$$

This work constitute an extension of the work presented for linear and semidefinite optimization by the same authors L. Derbal et al. in [8], [9].

Since $P_*(\kappa)$-LCP is a generalization of LO, we lose the orthogonality of the search direction dx and ds.Therefore, the analysis is more complicated than the linear optimization case.

With this kernel function new search directions and the proximity function to measure the distance between the iterate and the center path are determined for the proposed algorithm.

We prove that the proposed interior-point algorithm has the same complexity bound as the best known interior-point algorithms for solving these types of problems. Using some conditions, we show that the worst-case iteration complexity of the new algorithm is

$$O((1+2\kappa)\,q\,\sqrt{n}(\log\sqrt{n})^{\frac{q+1}{q}}\log\frac{n}{\epsilon}).$$

which becomes $O((1+2\kappa)\sqrt{n}\log n\log\frac{n}{\epsilon})$ with a special choice of the parameter q. It is currently the best known result for large-update primal-dual IPM.

Furthermore, by providing numerical results we also show the efficiency of the proposed interior-point algorithm (IPA). We implemented the theoretical version of the IPA. We compared our IPA to the interior-point algorithm using the function $\psi_1(t) = \left(\frac{t^2-1}{2} + \frac{4}{\pi p}\left[\tan^p\left(\frac{\pi}{2t+2}\right) - 1\right], p \geq 2\right)$ proposed in [10].

In the theoretical analysis, the step size is usually given a value that is very small during each inner iteration. In practice, this leads to very large inner iteration number. So, to accelerate the iteration process we propose a dynamic and practical choices for the step size [8].

The paper is organized as follows. In the next section, we give some basic concepts and useful results about the $P_*(\kappa)$-LCP and $P_*(\kappa)$-matrices. Furthermore, we decribe the primal-dual interior point method for solving the LCP. The new IPA for $P_*(\kappa)$-LCP is introduced, which is based on a new search directions by using the new kernel function. In Section 3, we present the new kernel function and its properties. In Section 4, we evaluate a default step size which not only keeps the iterates feasible but also leads the barrier function to have a sufficiently large decrease of the value in each inner iteration. Section 5 contains the complexity analysis of the introduced IPA with the new search directions. While, in Section 6 numerical computations are presented and compared to the computational performance of an other kernel function introduced in 2017 by M. El Ghami and Wang, [10]. In Section 7, some concluding remarks are presented.

The following notations are used throughout the paper. The nonnegative orthant and positive orthant are denoted as $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$, respectively. For $x = (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n$, $x_{\min}$ denotes the smallest value of the components of x, X is the diagonal matrix whose diagonal elements are the coordinates of vector x, i.e., $X = diag(x)$. e denotes the all-one vector of length n, i.e., $e = (1, 1, ..., 1)^T$. The index set I is $I = \{1, 2, ..., n\}$. Furthermore, $\|x\| = \sqrt{x^T x}$ denotes the 2-norm of the vector x. $x^T y = \sum_{i=1}^{n} x_i y_i$ for $x, y \in \mathbb{R}^n$. Finally, we say $f(t) = O(g(t))$ if there exists a positive constant ω such that $f(t) \leq \omega g(t)$ holds for all $t > 0$, and $f(t) = \Theta(g(t))$ if there exists some positive constants ω_1 and ω_2 such that $\omega_1 g(t) \leq f(t) \leq \omega_2 g(t)$ holds for all $t > 0$, where f and g are two positive real-valued functions.

2. Preliminaries

In this section we introduce the definition of $P_*(\kappa)$-matrix and its properties [14].

Definition 2.1. *[14] Let $\kappa \geq 0$ be a nonnegative number. A matrix $M \in \mathbb{R}^{n\times n}$ is called a $P_*(\kappa)$-matrix if*

$$(1 + 4\kappa) \sum_{i\in I_+(x)} x_i (Mx)_i + \sum_{i\in I_-(x)} x_i (Mx)_i \geq 0,$$

for all $x \in \mathbb{R}^n$, where

$$I_+(x) = \{i \in I : x_i (Mx)_i \geq 0\}, \qquad I_-(x) = \{i \in I : x_i (Mx)_i < 0\}.$$

The class of all $P_*(\kappa)$-matrices is denoted by $P_*(\kappa)$, and the class P_* is defined by $P_* = \bigcup_{\kappa\geq 0} P_*(\kappa)$, i.e., M is a P_* -matrix if $M \in P_*(\kappa)$ for some $\kappa \geq 0$. Obviously, $P_*(0)$ is the class of positive semidefinite matrices.

Proposition 2.1. *[14] If $M \in \mathbb{R}^{n\times n}$ is a $P_*(\kappa)$-matrix, then*

$$\bar{M} = \begin{pmatrix} -M & I \\ S & X \end{pmatrix}$$

is a nonsingular matrix for any positive diagonal matrices $X, S \in \mathbb{R}^{n\times n}$.

The following corollary is used to prove that the modified Newton-system has a unique solution.

Corollary 2.1. *[14] Let $M \in \mathbb{R}^{n\times n}$ be a $P_*(\kappa)$- matrix and $x, s \in \mathbb{R}^n_{++}$.Then for all $\omega \in \mathbb{R}^n$ the system*

$$\begin{cases} -M\Delta x + \Delta s = 0 \\ S\Delta x + X\Delta s = \omega, \end{cases}$$

has a unique solution $(\Delta x, \Delta s)$, where X and S are the diagonal matrices obtained from the vectors x and s.

The basic idea of IPMs for $P_*(\kappa)$-LCP is to replace the second equation in $P_*(\kappa)$-LCP by the parameterized equation $xs = \mu e$, $\mu > 0$. This replacement leads us to consider the following parameterized system:

$$\begin{cases} s = Mx + q, \\ xs = \mu e, \\ x \geq 0,\ s \geq 0. \end{cases} \tag{2.1}$$

Without loss of generality, we assume that (1.1) satisfies the interior point condition (IPC), i.e., there exists $(x_0, s_0) > 0$ such that $s_0 = Mx_0 + q$. Since M is a $P_*(\kappa)$-matrix and (1.1) satisfies the IPC, the parameterized system (2.1) has a unique solution for any $\mu > 0$ (Lemma 4.3 of [14]) and it is denoted as $\left(x_\mu, s_\mu\right)$. We call it μ-center for $\mu > 0$ and the solution set $\left\{\left(x_\mu, s_\mu\right) \setminus \mu > 0\right\}$ is called the central path of (1.1). As goes to zero, the limit of the central path exists and it naturally yields the optimal solution for (1.1) (Theorem 4.4 of [14]). Moreover, assuming that we have an initial point $(x, s) := (x_0, s_0)$ which is in a certain neighborhood of some μ-center. Then we decrease μ to $\mu := (1 - \theta)\mu$ for some $\theta \in (0, 1)$. Using Newton's method to the system (2.1), we have

$$\begin{cases} -M\Delta x + \Delta s = 0 \\ S\Delta x + X\Delta s = \mu e - xs. \end{cases} \tag{2.2}$$

According to Corollary 1, we get the unique search direction $(\Delta x, \Delta s)$. This direction approximates the next μ-center. By taking a step along the search direction $(\Delta x, \Delta s)$, we get a new iteration (x^+, s^+), where

$$x^+ := x + \alpha\Delta x, \quad s^+ := s + \alpha\Delta s,$$

for some step size $0 < \alpha \leq 1$.

For the formulation and analysis of the generic interior point method for the $P_*(\kappa)$-LCP, the introduction of the following vectors is critical:

$$v = \sqrt{\frac{xs}{\mu}},\ p = \sqrt{\frac{x}{s}},\ d_x := \frac{v\Delta x}{x} \text{ and } d_s := \frac{v\Delta s}{s}. \tag{2.3}$$

Then we have the scaled Newton system as follows

$$\begin{cases} -\overline{M}d_x + d_s = 0 \\ d_x + d_s = v^{-1} - v, \end{cases} \tag{2.4}$$

where $\overline{M} := PMP$ with $P = diag(p)$. Note that the second equation in (2.4) is called the scaled centering equation, which equals the negative gradient of the classical logarithmic barrier function $\Psi_c(v)$, i.e.,

$$d_x + d_s = -\nabla\Psi_c(v), \tag{2.5}$$

where $\Psi_c(v) := \sum_{i=1}^{n} \psi_c(v_i)$, $\psi_c(t) = \dfrac{t^2 - 1}{2} - \log t$, $t > 0$.

We call $\psi_c(t)$ the kernel function of the classical logarithmic proximity function $\Psi_c(v)$. The key idea in the new variant of IPMs based on kernel functions is to replace $\Psi_c(v)$ by a proximity function $\Psi(v)$, such that

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

Hence, the value of $\Psi(v)$ can be considered as a proximity measure for closeness with respect to the μ-center $\left(x_\mu, s_\mu\right)$. In what follows, we define the norm-based proximity measure $\delta(v)$

$$\delta(v) := \frac{1}{2}\|\nabla\Psi(v)\| = \frac{1}{2}\|d_x + d_s\|. \tag{2.6}$$

We can easily verify that

$$\delta(v) = 0 \Leftrightarrow v = e \Leftrightarrow xs = \mu e.$$

From the solution d_x and d_s, the vectors Δx and Δs can be calculated using (2.3). The algorithm works as follows. Suppose that we give a strictly feasible point (x_0, s_0) in a τ-neighborhood of the current μ-center, i.e., $\Psi(v) < \tau$. Then the value of μ is reduced by the factor $1 - \theta$ with some fixed $\theta \in (0, 1)$, which changes the value of v and obtains a new μ-center $\left(x_\mu, s_\mu\right)$. Hence the value of the proximity function will likely exceed the threshold value of τ , i.e., $\Psi(v) \geq \tau$. Now we start the inner iteration by solving the system (2.4) and (2.3) to obtain the unique search direction. In order to reduce the value of the proximity function $\Psi(v)$, the step size α should choice appropriately. If necessary, we repeat this process until we find the iterate that again belongs to the τ -neighborhood of the corresponding μ-center, i.e., $\Psi(v) < \tau$. Next, a new outer iteration starts by reducing the value of μ again. This procedure is repeated until μ is small enough, say until $n\mu \leq \epsilon$.

The algorithm1 considered in this paper is described in Figure 1.

Generic interior-point algorithm for $P_*(\kappa)$-LCP

Input:
A proximity function $\Psi(v)$;
a threshold parameter $\epsilon > 0$;
a barrier update parameter θ, $0 < \theta < 1$;
a strictly feasible point (x_0, s_0) and $\mu_0 = 1$ such that $\Psi\left(\sqrt{x_0 s_0 / \mu_0}\right) < \tau$;
begin
  $x := x_0$; $s := s_0$; $\mu = \mu_0$;
  while $n\mu > \epsilon$ do
    $\mu := (1 - \theta)\mu$; $v := \sqrt{\frac{xs}{\mu}}$;
    while $\Psi(v) \geq \tau$ do
      find search directions (d_x, d_s) by solving system (2.4)
      compute $(\Delta x, \Delta s)$ from (2.2)
      determine a step size α;
      update $x := x + \alpha\Delta x$; $s := s + \alpha\Delta s$;
    end while
  end while
end.

Figure1. Generic algorithm.

Remark 2.1. *If $\tau = O(n)$ and $\theta = \Theta(1)$, then the method is called a large-update method. When $\tau = O(1)$ and $\theta = \Theta(\frac{1}{\sqrt{n}})$, we call the method a small-update method.*

3. Properties of the new kernel function

In this section,we study the basic properties of $\psi(t)$ and $\Psi(v)$. For ease of reference, we give the first three derivatives of $\psi(t)$ with respect to t as follows

$$\begin{cases} \psi'(t) = t - \dfrac{1}{2t} - \dfrac{e^{\frac{1}{t^q}-1}}{2t^{q+1}}, \\ \psi''(t) = 1 + \dfrac{1}{2t^2} + \dfrac{1}{2}\left(\dfrac{(q+1)t^q + q}{t^{2q+2}}\right)e^{\frac{1}{t^q}-1}, \\ \psi'''(t) = \dfrac{-1}{t^3} - \dfrac{1}{2}(q^2t^{-(3q+3)} + 3q(q+1)t^{-(2q+3)} + (q+1)(q+2)t^{-(q+3)})e^{\frac{1}{t^q}-1}. \end{cases} \tag{3.1}$$

It is quite straightforward to verify

$$\begin{cases} \psi(1) = 0, \psi'(1) = 0; \\ \psi''(t) > 0,\ \psi'''(t) < 0, t > 0; \\ \lim\limits_{t\to 0^+}\psi(t) = \lim\limits_{t\to+\infty}\psi(t) = +\infty. \end{cases} \tag{3.2}$$

Moreover, from (3.2) $\psi(t)$ is strictly convex and $\psi''(t)$ is monotonically decreasing in $t \in (0, \infty)$.
The following lemmas are used to prove the eligibility of the kernel function.

Lemma 3.1. *[8] For $\psi(t)$, we have the following*
$\psi(t)$ is exponentially convex for all $t > 0$; that is
(a) $\psi\left(\sqrt{t_1t_2}\right) \le \dfrac{1}{2}\left(\psi(t_1) + \psi(t_2)\right)$,
(b) $\psi''(t)$ is monotonically decreasing for all $t > 0$,
(c) $t\psi''(t) - \psi'(t) > 0$ for all $t > 0$,
(d) $\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, t > 1, \beta > 1$.

Lemma 3.2. *[8] For $\psi(t)$, we have*

$$\frac{1}{2}(t-1)^2 \le \psi(t) \le \frac{1}{2}\left[\psi'(t)\right]^2, \quad t > 0 \tag{3.3}$$

$$\psi(t) \le \frac{2+q}{2}(t-1)^2, \quad t > 1. \tag{3.4}$$

Let $\psi(t)$ be as defined in (1.3), one has

$$t\psi'(t) \ge \psi(t), \quad t > 1.$$

Let $\varrho : [0, \infty) \to [1, \infty)$ be the inverse function of $\psi(t)$ for $t \ge 1$ and $\rho : [0, +\infty[\to]0, 1]$ be the inverse function of $-\dfrac{1}{2}\psi'(t)$ for all $t \in]0, 1]$. In the next lemma we use the so-called barrier term $\psi_b(t)$ of $\psi(t)$, which is defined by

$$\psi(t) = \frac{t^2-1}{2} + \psi_b(t), \quad t > 0.$$

$\rho : [0, \infty) \to (0, 1]$ be the inverse function of the restriction of $-\psi_b'(t)$ in the interval $(0, 1]$ and $s_b = -\psi_b'(t)$. Then one has

Lemma 3.3. *[8] For $\psi(t)$, we have*

$$1 + \sqrt{\frac{2s}{q+2}} \leq \varrho(s) \leq 1 + \sqrt{2s}. \tag{3.5}$$

$$\rho(s) \geq \underline{\rho}(1+2s), \tag{3.6}$$

$$\underline{\rho}(s_b) \geq \frac{1}{(\log(2s_b)+1)^{\frac{1}{q}}}.s_b > \frac{1}{2}. \tag{3.7}$$

Lemma 3.4. *[8] Let δ be defined in* (2.6)**.** *So, we get :*

$$\delta(v) \geq \sqrt{\frac{1}{2}\Psi(v)}. \tag{3.8}$$

Remark 3.1. *In the following, we assume that $\Psi(v) \geq \tau \geq 1$. According to the previous lemma we have: $\delta(v) \geq \frac{1}{\sqrt{2}}$.*

3.1. 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 start with an important theorem which is valid for all kernel functions ψ which are strictly convex, and satisfy Lemma 3.1 (**c**).

Theorem 3.1. *[14] Let $\varrho : [0, +\infty) \to [1, +\infty)$ be the inverse function of $\psi(t), t \geq 1$. Then for any positive vector v and any $\beta \geq 1$,*

$$\Psi(\beta v) \leq n\psi(\beta\varrho(\frac{\Psi(v)}{n})).$$

Proof. Using Lemma 3.1(d), and Theorem 3.2 in [2], we can get the result. This completes the proof. □

Lemma 3.5. *Let $0 \leq \theta \leq 1, v_+ = \frac{v}{\sqrt{1-\theta}}$. If $\Psi(v) \leq \tau$, then*

$$\Psi(v_+) \leq \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)}.$$

Proof. Since $\frac{1}{\sqrt{1-\theta}} \geq 1$ and $\varrho(\frac{\Psi(v)}{n}) \geq 1$, then $\frac{\varrho(\frac{\Psi(v)}{n})}{\sqrt{1-\theta}} \geq 1$, and for $t \geq 1$, we have $\psi(t) \leq \frac{t^2-1}{2}$.

Using Theorem 3.1 *with* $\beta = \frac{1}{\sqrt{1-\theta}}$, (3.5), *and* $\Psi(v) \leq \tau$, *we have*

$$\begin{aligned}
\Psi(v_+) &\leq n\psi\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right)\right) \\
&\leq \frac{n}{2}\left(\left[\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{n}\right)\right]^2 - 1\right) \\
&\leq \frac{n}{2(1-\theta)}\left(\left[1+\sqrt{2\frac{\Psi(v)}{n}}\right]^2 - (1-\theta)\right) \\
&= \frac{n}{2(1-\theta)}\left(\left[1+2\frac{\Psi(v)}{n}+2\sqrt{2\frac{\Psi(v)}{n}}\right] - (1-\theta)\right) \\
&\leq \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)}.
\end{aligned}$$

□

Denote

$$\Psi_0 = \frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)} = L(n,\theta,\tau), \tag{3.9}$$

as an upper bound of Ψ, during the process of the algorithm.

4. A default value for the step size

In this section, we evaluate a default step size which not only keeps the iterates feasible but also leads the barrier function to have a sufficiently large decrease of the value in each inner iteration. Since $P_*(\kappa)$-LCP is generalization of LO, we lose the orthogonality of the search direction d_x and d_s. Therefore, the analysis is more complicated than the LO case.

After a step with size α the new iterates are $x_+ = x + \alpha\Delta x$, $s_+ = s + \alpha\Delta s$.

Using (2.3), we get

$$\begin{cases} x_+ = x\left(e + \alpha\frac{\Delta x}{x}\right) = x\left(e + \alpha\frac{dx}{v}\right) = \frac{x}{v}(v + \alpha d_x), \\ s_+ = s\left(e + \alpha\frac{\Delta s}{s}\right) = x\left(e + \alpha\frac{ds}{v}\right) = \frac{s}{v}(v + \alpha d_s). \end{cases}$$

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

$$v_+ = \sqrt{\frac{x_+ s_+}{\mu}} = \sqrt{(v + \alpha d_x)(v + \alpha d_s)},$$

then

$$v_+^2 = \frac{x_+ s_+}{\mu} = (v + \alpha d_x)(v + \alpha d_s).$$

By Lemma 3.1(a), one easily verifies that

$$\Psi(v_+) = \Psi(\sqrt{(v + \alpha d_x)(v + \alpha d_s)}) \leq \frac{1}{2}(\Psi(v + \alpha d_x) + \Psi(v + \alpha d_s)).$$

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

$$f_1(\alpha) = \frac{1}{2}(\Psi(v + \alpha d_x) + \Psi(v + \alpha d_s)) - \Psi(v).$$

Obviously,

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

Taking the first two derivatives of $f_1(\alpha)$ with respect to α, we have

$$\begin{aligned} f_1^{'}(\alpha) &= \frac{1}{2}\sum_{i=1}^{n}(\psi'(v_i + \alpha d_{xi})d_{xi} + \psi'(v_i + \alpha d_{si})d_{si}), \\ f_1^{''}(\alpha) &= \frac{1}{2}\sum_{i=1}^{n}(\psi''(v_i + \alpha d_{xi})d_{xi}^2 + \psi''(v_i + \alpha d_{si})d_{si}^2). \end{aligned}$$

Using (2.5) and (2.6), we have

$$f_1^{'}(0) = \frac{1}{2}\nabla\Psi(v)^T(d_x + d_s) = -\frac{1}{2}\nabla\Psi(v)^T\nabla\Psi(v) = -2\delta(v)^2.$$

For convenience, we denote

$$\delta := \delta(v),\ \sigma_+ := \sum_{i\in I_+} d_{x_i}d_{s_i},\ \sigma_- := -\sum_{i\in I_-} d_{x_i}d_{s_i}.$$

Since M is a $P_*(\kappa)-$matrix and $M\Delta x = \Delta s$, we have

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

where $I_+ = \{i \in I : \Delta x_i\Delta s_i \geq 0\}$, and $I_- = I - I_+$. Since

$$d_x d_s = \frac{v^2\Delta x\Delta s}{xs} = \frac{\Delta x\Delta s}{\mu},$$

and $\mu > 0$, hence

$$(1 + 4\kappa)\sum_{i\in I_+} d_{x_i}d_{s_i} + \sum_{i\in I_-} d_{x_i}d_{s_i} = (1 + 4\kappa)\sigma_+ - \sigma_- \geq 0. \tag{4.1}$$

The following technical lemmas give the upper bound of σ_+, σ_-, $\|d_x\|$ and $\|d_s\|$, respectively.
From Lemmas 4.1−4.2 in [6]. We have the following Lemmas 4.1−4.2.

Lemma 4.1. *$\sigma_+ \leq \delta^2$ and $\sigma_- \leq (1 + 4\kappa)\delta^2$.*

Lemma 4.2. *$\sum_{i=1}^{n}\left(d_{x_i}^2 + d_{s_i}^2\right) \leq 4(1 + 4\kappa)\delta^2, \| d_x \| \leq 2\sqrt{1 + 2\kappa}\delta$ and $\| d_s \| \leq 2\sqrt{1 + 2\kappa}\delta$.*

The subsequent lemmas lead to obtaining the default step size. Using the above Lemmas 4.1, 4.2 and our kernel function, their proofs are easy modification of the similar Lemmas 4.3−4.4 stated in [6].

Lemma 4.3. $f_1''(\alpha) \leq 2\ (1+2\kappa)\,\delta^2\psi''\left(v_{\min} - 2\alpha\sqrt{1+2\kappa}\delta\right)$.

Lemma 4.4. $f_1'(\alpha) \leq 0$ *if* α *is such that*

$$-\psi'\left(v_{\min} - 2\alpha\sqrt{1+2\kappa}\delta\right) + \psi'(v_{\min}) \leq \frac{2\,\delta}{\sqrt{1+2\kappa}}. \tag{4.2}$$

From Lemmas 4.5 − 4.6 in [6], we have the following lemmas 4.5, 4.6 gives an upper bound of (α) in terms of δ and ψ''.

Lemma 4.5. *Let* $\rho : [0, +\infty) \to (0, 1]$ *be the inverse function of* $-\frac{1}{2}\psi'(t)$ *for all* $t \in (0, 1]$. *Then the largest step size* $\bar{\alpha}$ *satisfying (4.2) is given by*

$$\overline{\alpha} := \frac{1}{2\,\delta\sqrt{1+2\kappa}}\left[\rho(\delta) - \rho\left(\left(1 + \frac{1}{\sqrt{1+2\kappa}}\right)\delta\right)\right], \tag{4.3}$$

Lemma 4.6. *Let* ρ *and* $\bar{\alpha}$ *be as defined in Lemma* 4.5. *then*

$$\bar{\alpha} \geq \frac{1}{(1+2\kappa)\,\psi''(\rho((1+\frac{1}{\sqrt{1+2\kappa}})\delta))}. \tag{4.4}$$

Lemma 4.7. *Let* ρ *and* $\bar{\alpha}$ *be as defined in Lemma* 4.5. *If* $\Psi(v) \geq \tau \geq 1$, *then we have*

$$\bar{\alpha} \geq \frac{1}{(1+2\kappa)\left(1 + (2q+1)(1+4\delta)\left[\log(2+8\delta)+1\right]^{\frac{q+1}{q}}\right)},$$

Proof. Using Lemma 4.6 and (3.6), we get

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

Hence, putting $t = \underline{\rho}(1+2(1+\frac{1}{\sqrt{1+2\kappa}})\delta)$, $0 < t \leq 1$, it follows that

$$\begin{aligned}
\bar{\alpha} &\geq \frac{1}{(1+2\kappa)\left(1 + \frac{1}{2t^2} + \left[\frac{1}{2}(q+1)t^{-(q+2)} + \frac{1}{2}qt^{-(2q+2)}\right]e^{t^{-q}-1}\right)} \\
&= \frac{1}{(1+2\kappa)\left(1 + \frac{1}{2t^2} + \left[(q+1)t^{-1} + qt^{-(q+1)}\right](-\psi_b'(t) - \frac{1}{2t}\right)} \\
&= \frac{1}{(1+2\kappa)\left(1 - \frac{q}{2t^2}(1+t^{-q}) + \left[(q+1)t^{-1} + qt^{-(q+1)}\right](-\psi_b'(t))\right)} \\
&> \frac{1}{(1+2\kappa)\left(1 + (2q+1)t^{-(q+1)}(-\psi_b'(t))\right)} \\
&> \frac{1}{(1+2\kappa)\left(1 + (2q+1)(1+2(1+\frac{1}{\sqrt{1+2\kappa}})\delta)\left[\log(2+4(1+\frac{1}{\sqrt{1+2\kappa}})\delta)+1\right]^{\frac{q+1}{q}}\right)}.
\end{aligned}$$

Using that $1 + \frac{1}{\sqrt{1+2\kappa}} \leq 2$ for all $\kappa \geq 0$, we get

$$\bar{\alpha} > \frac{1}{(1+2\kappa)\left(1 + (2q+1)(1+4\delta)\left[\log(2+8\delta)+1\right]^{\frac{q+1}{q}}\right)}.$$

□

Denoting

$$\tilde{\alpha} = \frac{1}{(1+2\kappa)\left(1 + (2q+1)(1+4\delta)\left[\log(2+8\delta)+1\right]^{\frac{q+1}{q}}\right)}, \tag{4.5}$$

we have that $\tilde{\alpha}$ is the default step size and that $\tilde{\alpha} \leq \bar{\alpha}$.

5. Iteration complexity

In the present section, we derive worst-case iteration complexity of Algorithm 1. The following result serves to get an estimate value of $f(\tilde{\alpha})$, for its proof we refer to Lemma 4.5 in [2].

Lemma 5.1. *If the step size α is such that $\alpha \leq \bar{\alpha}$, then $f(\alpha) \leq -\alpha\delta^2$.*

Theorem 5.1. *Let ρ be as defined in Lemma* 4.5, *$\tilde{\alpha}$ as defined in* (4.5) *and $\Psi(v) \geq 1$. Then*

$$f(\tilde{\alpha}) \leq -\frac{\delta^2}{(1+2\kappa)\left(1 + (2q+1)(1+4\delta)\left[\log(2+8\delta)+1\right]^{\frac{q+1}{q}}\right)} \tag{5.1}$$

$$\leq -\frac{\sqrt{\Psi(v)}}{(1+2\kappa)\left(2 + (2q+1)(1+4\sqrt{2})\left[\log(2+4\sqrt{2\Psi_0})+1\right]^{\frac{q+1}{q}}\right)} \tag{5.2}$$

Proof. Using Lemma 5.1 and $\tilde{\alpha}$ defined in(4.5), the first inequality follows. Since the right hand side of expression in (5.1) is monotonically decreasing in δ. The last inequality follows from (5.1) and $\Psi_0 \geq \Psi(v) \geq \tau \geq 1$. This result holds the theorem. □

Lemma 5.2. *(Lemma* 14 *in [17]) Let $t_0, t_1, ..., t_k$ be a sequence of positive number such that*

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

where $\beta > 0$, and $0 < \gamma \leq 1$. Then $K \leq \left[\frac{t_0^{\gamma}}{\beta\gamma}\right]$.

Lemma 5.3. *If K denotes the number of inner iterations in the outer iterations. Then we have*

$$K \leq (1+2\kappa)\left(4+(2q+1)(4+8\sqrt{2})\left[\log(2+4\sqrt{2\Psi_0})+1\right]^{\frac{q+1}{q}}\right)\Psi_0^{\frac{1}{2}}.$$

Proof. The definition of K implies $\Psi_{K-1}(v) > \tau$ and $\Psi_K(v) \leq \tau$. According to Theorem 5.1, we obtain that the sequence $\Psi_k(v)$ satisfies

$$\Psi_{k+1}(v) \leq \Psi_k(v) - \frac{\sqrt{\Psi_k(v)}}{(1+2\kappa)\left(2+(2q+1)(1+4\sqrt{2})\left[\log(2+4\sqrt{2\Psi_0})+1\right]^{\frac{q+1}{q}}\right)},$$

by Lemma 5.2, we can find the appropriate values of β , t_0, and γ

$$\beta = \frac{1}{2+(2q+1)(2+4\sqrt{2})\left[\log(2+4\sqrt{2\Psi_0})+1\right]^{\frac{q+1}{q}}}, \gamma = \frac{1}{2}, t_0 = \Psi_0.$$

This completes the proof. □

Theorem 5.2. *Given that $\tau = O(\sqrt{n})$ and $\theta = \Theta(1)$, which are characteristics of large-update methods, Algorithm 1 will obtain an ϵ-approximate solution of $P_*(\kappa)$-LCP in at most $O((1+2\kappa)\, q\, \sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log\frac{n}{\epsilon})$ iterations.*

Proof. It is well known that the number of outer iterations is bounded above by $\frac{1}{\theta}\log\frac{n}{\epsilon}$(Theorem 5.4 of [4]). By multiplying this number with the upper bound for the number of inner iterations per outer iteration, We get the upper bound for the total number of iterations, namely

$$\frac{K}{\theta}\log\frac{n}{\varepsilon} \leq (1+2\kappa)\left(4+(2q+1)(4+8\sqrt{2})\left[\log(2+4\sqrt{2\Psi_0})+1\right]^{\frac{q+1}{q}}\right)\Psi_0^{\frac{1}{2}}\log\frac{n}{\epsilon}.$$

For large-update methods with $\tau = O(\sqrt{n})$ and $\theta = \Theta(1)$, we have $\Psi_0 = O(n)$ and $O((1+2\kappa)\, q\, \sqrt{n}(\log \sqrt{n})^{\frac{q+1}{q}} \log\frac{n}{\epsilon})$ iterations complexity. □

Remark 5.1. *Note that the iteration bound related to the selection of parameter q. Especially, if taking $q = \log\log\sqrt{n}$, we have the complexity $O((1+2\kappa)\ \sqrt{n}\log n\log\frac{n}{\epsilon})$ for $P_*(\kappa)$-LCP. This is matches the currently best known iteration bound for $P_*(\kappa)$-LCP.*

6. Numerical Results

We present in this chapter some comparative numerical tests between a kernel function studied recently in 2017 defined in [10] by:

$$\psi_1(t) = \left(\frac{t^2-1}{2} + \frac{4}{\pi p}\left[\tan^p\left(\frac{\pi}{2t+2}\right)-1\right],\ p \geq 2\right)$$

with $O\left((1+2\kappa)\,pn^{\frac{p+2}{2(p+1)}}\log\frac{n}{\epsilon}\right)$ complexity results that correspond to large-update methods, in order to examine the influence of the choice of the new kernel function on the behavior of Algorithm 1. The algorithm is coded in MATLAB (R2009b) and executed on a PC with 2.40 GHz processor speed. We have taken: $\epsilon = 10^{-6}$, $\theta = 0,99$ and $\tau = 10$ for examples of fixed size and $\tau = n$ for examples with a variable size. The choice of the step size α, $(0 < \alpha \leq 1)$ is another crucial issue in the analysis of the algorithm. In the theoretical analysis, the step size α is usually given a value that is very small during each inner iteration. In practice, this leads to very large inner iteration number. So, to accelerate the iteration process we propose a dynamic and practical choices defined bellow:

Dynamic choice: [20] We take $\alpha = p\tilde{\alpha}$, when $p \geq 1$ is a fixed scalar according to the the size of the increment of x or s and $\tilde{\alpha}$ is the default step size (the theoretical choice). In our numerical tests, we set:

$$\alpha = \begin{cases} p_1\tilde{\alpha} & \text{if } \|\Delta x\| \geq n \\ p_2\tilde{\alpha} & \text{if } 1 \leq \|\Delta x\| \leq n \\ p_3\tilde{\alpha} & \text{if } \|\Delta x\| \leq 1. \end{cases}$$

Practical choice : It should be noted that the step size selected during each inner iteration is small enough for analyzing the algorithm, while in practice the step size during each inner iteration should be large enough for the efficiency of the algorithm. Then the step sizes α_x and α_s during each inner iteration in this experiment are chosen according to the following strategy. First, compute the maximum allowable step sizes by the following strategy:

$$\alpha_x = \beta\left\{1, \min_{i\in I}(-\frac{x_i}{dx_i})\right\}\ ,\quad \alpha_s = \beta\left\{1, \min_{i\in J}(-\frac{s_j}{ds_j})\right\},$$

such as $0 < \beta < 1$, $I = \{i : dx_i < 0\}$ and $J = \left\{j : ds_j < 0\right\}$.

In this paper, we choose $\beta \in \{0.6, 0.9, 0.98, 0.995\}$. The new iteration point is defined by

$$x := x + \alpha_x\Delta x\ ; \quad s := s + \alpha_s\Delta s$$

Theoretical choice:

For our kernel function ψ, we take the default step-size $\tilde{\alpha}$ defined in the formula (4.5) and for the kernel function ψ_1, the default step-size is defined below:

$$\begin{aligned} \alpha &= \frac{1}{(1+2\kappa)\,(9+4\pi p)\,(8\delta+2)^{\frac{p+2}{p+1}}} \\ &= \frac{1}{(1+2\kappa)\,(9+8\pi)\,(8\delta+2)^{\frac{4}{3}}}\ ,\quad (p=2) \end{aligned}$$

6.1. Examples of fixed size

We consider the following $p_*(0)$-LCP:

Example 6.1.

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix} \text{ and } q = \begin{pmatrix} -8 \\ -6 \\ -4 \\ 3 \end{pmatrix}.$$

The starting point is:

$$(x_0)^T = \begin{pmatrix} 1.5 & 0.4 & 0.2 & 7 \end{pmatrix}.$$

The solution is:

$$(x^*)^T = \begin{pmatrix} 2.5 & 0.5 & 0 & 2.5 \end{pmatrix}.$$

Example 6.2.

$$M = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix} \text{ and } q = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}.$$

The starting point is:

$$(x_0)^T = \begin{pmatrix} 0.65 & 0.65 & 0.65 & 0.65 & 0.65 & 0.65 & 0.65 \end{pmatrix}.$$

The solution is:

$$(x^*)^T = \begin{pmatrix} 0.3660, & 0.4639, & 0.4897 & 0.4948 & 0.4897 & 0.4639 & 0.3660 \end{pmatrix}.$$

Example 6.3. *We consider the following matrix:*

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 0.8 & 0.32 & 1.128 & 0.0512 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.32 & 0.128 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 032 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0512 & -0.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } q = \begin{pmatrix} -0.0256 \\ -0.064 \\ -0.16 \\ -0.4 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

The starting point is:

$$(x_0)^T = \begin{pmatrix} 0.18 & 0.18 & 0.18 & 0.18 & 0.25 & 3 & 4 & 5 & 6 & 9 \end{pmatrix}.$$

The solution is:

$$(x^*)^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

In the results tables, n represents the size of the example, (*Inner*) represents the number of inner iterations and (*Cpu*)) represents the calculation time in seconds. In our computational study, we compared our algorithm using the function $\psi(t) = \dfrac{t^2 - 1 - \log t}{2} + \dfrac{e^{\frac{1}{t^q}-1} - 1}{2q}$ $(q \geq 1)$ with the algorithm proposed in [10], which is based on the function

$$\psi_1(t) = \left(\frac{t^2-1}{2} + \frac{4}{\pi p}\left[\tan^p\left(\frac{\pi}{2t+2} \right) - 1 \right], p \geq 2 \right).$$

We summarize our results in the following tables:

Table 1: Theoretical choice of the step size α.

	ψ		ψ_1	
Exp	*Inner*	*Cpu*	*Inner*	*Cpu*
1	19416	12.955010	66055	36.528270
2	21739	13.521766	37124	31.188794
3	23920	33.156218	47813	97.466112

In this table 1, we have implemented the theoretical version of our proposed algorithm (Algorithm1) and the algorithm based on the kernel function ψ_1 proposed by El Ghami et al. on examples of fixed size.

Table 2: Dynamic choice of the step step size α.

	ψ		ψ_1	
Exp	*Inner*	*Cpu*	*Inner*	*Cpu*
1	1936	2.637886	6599	6.518613
2	2163	2.670526	3710	5.164342
3	2376	28.235506	4777	67.021835

Table 2 contains the number of iterations and times in case of the two aformentioned algorithms using the dynamic choice of the step size α.

In these two cases our algorithm (Algorithm1) provided better results in number of iterations and CPU times.

Table 3: Practical choice of the step size α.

	ψ		ψ_1	
Exp	*Inner*	*Cpu*	*Inner*	*Cpu*
Exp1/β = 0.995	2	0.078766	2	0.081290
Exp1/β = 0.98	3	0.104999	2	0.087865
Exp1/β = 0.9	3	0.082444	3	0.082052
Exp1/β = 0.6	6	0.081271	5	0.091480
Exp2/β = 0.995	2	0.007927	4	0.118633
Exp2/β = 0.98	2	0.006171	3	0.089546
Exp2/β = 0.9	3	0.081675	3	0.077614
Exp2/β = 0.6	6	0.084934	6	0.079696
Exp3/β = 0.995	1	0.126247	1	0.550030
Exp3/β = 0.98	2	0.087238	3	0.070101
Exp3/β = 0.9	3	0.065264	3	0.092191
Exp3/β = 0.6	6	0.100081	7	0.098438

Table 3 In this table, we present the results that are related to the two algorithms using the practical choice of the step size α with different values of the parameter β .

It can be observed that in these cases the two algorithmes using different values of β provide usually similar results.

6.2. Examples of variable size

We consider the following $p_*(0)$-LCP with

$$mij = \begin{cases} 1 & \text{si} \quad i = j \\ 2 & \text{si} \quad i < j \\ 0 & \text{si} \quad i > j \end{cases} \quad \text{and} \quad q = \begin{pmatrix} -1 \\ . \\ . \\ -1 \end{pmatrix} \quad \text{for} \begin{cases} 1 \le i \le n. \\ 1 \le j \le n. \end{cases}$$

The starting point is:

$$(x_0)^T = \begin{pmatrix} 0.05, & ..., & 0.05, & 1.05 \end{pmatrix}.$$

The solution is:

$$(x^*)^T = \begin{pmatrix} 0, & ..., & 0, & 1 \end{pmatrix}.$$

We have the results below:

Table 4: Theoretical choice of the step size α.

	ψ		ψ_1	
n	*Inner*	*Cpu*	*Inner*	*Cpu*
7	21887	16.328076	39456	52.682240
15	27006	30.057157	60968	80.536014
20	29741	88.326291	72260	105.006885
25	32223	412.491091	826353	926.069324
50	42714	864.884538	126763	2173.910673
75	51385	1767.782094	–	–

In table 4, we present numerical results of the theoretical version of Algorithm1 and the algorithm proposed by El Ghami et al. on examples of variable size. The obtained results are in favor with our algorithm mainly when the size of the example n exceeds 50 (the algorithm proposed in [10] fails).

Table 5: Dynamic choice of the step size α.

	ψ		ψ_1	
n	*Inner*	*Cpu*	*Inner*	*Cpu*
7	2190	2.666289	3937	4.468487
15	2699	6.314768	6083	30.590187
20	2952	25.132520	7226	74.879697
25	3207	41.092595	8252	15.425989
50	4262	89.855016	12658	214.390439
75	5131	125.561089	16375	392.228273

Table 5 contains the number of iterations and times in case of the two algorithms using the dynamic choice of the step size α. In this case, we can see that Algorithm1 provided better results.

Remark 6.1. *There is a parameter β involved in the definition of the practical choice, we used several values of this parameter as indicated above. These values were chosen after some preliminary experiments that showed that these values gave the most promising iteration counts. The value $\beta = 0.995$ gives the lowest iteration count in all cases.*

Table 6: Practical choice of the step size α.

	$\psi, (\beta = 0.995)$	
n	*Inner*	*Cpu*
7	2	0.004540
15	2	0.006865
30	2	0.169280
50	2	0.212657
75	2	0.229522
100	2	0.294217
150	2	0.525282
300	2	3.454890

In this table 6, we present the results that are related to Algorithm1 using the practical choice of the step size α with $\beta = 0.995$.

we obtained promising results when using the practical choice of the step size.

6.3. Example of a nonmonotonic LCP

Example 6.4. *We consider a $P_*(\kappa)$-LCP, with*

$$M = \begin{pmatrix} 0 & 1+4\kappa & 0 \\ -1 & 0 & 0 \\ 0 & 0 & c \end{pmatrix} \text{ and } q = \begin{pmatrix} 0.01 \\ 0.501 \\ -0.49 \end{pmatrix} \text{ with } c = 1 \text{ and } \kappa = 0, 0.5, 0.9.$$

The starting point is:

$$\begin{cases} \kappa = 0 \quad \text{we have } (x_0)^T = \begin{pmatrix} 0.2506, & 0.0674, & 0.5284 \end{pmatrix} \\ \kappa = 0.5 \text{ we have } (x_0)^T = \begin{pmatrix} 0.2506, & 0.0323, & 0.5260 \end{pmatrix} \\ \kappa = 0.9 \text{ we have } (x_0)^T = \begin{pmatrix} 0.2506, & 0.0220, & 0.5241 \end{pmatrix} \end{cases}$$

Table 7: Theoretical choice of the step size α.

	ψ		ψ_1	
κ	*Inner*	*Cpu*	*Inner*	*Cpu*
0	18594	8.770145	62837	28.334632
0.5	37048	17.077971	112711	43.963327
0.9	51713	23.587909	113218	54.193052

Table 7, represents the theoretical version of Algorithm1 and the algorithm proposed by El Ghami et al. on nonmonotonic LCP example with different values of κ. For this case,The results obtained are in favor with our algorithm.

Table 8: Dynamic choice of the step size α.

	ψ		ψ_1	
κ	*Inner*	*Cpu*	*Inner*	*Cpu*
0	1856	1.382202	6278	4.048021
0.5	3693	2.604641	11259	5.986296
0.9	5163	3.434282	11262	7.124869

In table 8, we compare our algorithm to the algorithm proposed in [10] applying to a nonmonotonic LCP example using the dynamic choice of the step size .

We observe that Algorithm1 provided better results in number of iterations and times.

Table 9: Practical choice of the step size α.

	ψ		ψ_1	
κ	*Inner*	*Cpu*	*Inner*	*Cpu*
0	2	0.113189	2	0.341297
0.5	2	0.083952	2	0.078926
0.9	2	0.012251	2	0.039677

Table 9 Here, we present the results that are related to the algorithms previously mentioned using the practical choice of the step size with different value of κ. We obtained promising results when using the practical choice of the step size. It seems that the practical choice is significantly better than the theoretical and dynamic choices.

Remark 6.2. *The results of these nine tables show that the algorithm based on our new kernel function ψ is efficient and that the number of iterations of the algorithm depends on the values of the step size α. It was noted that the theoretical value of the step size α is very small in each iteration, which requires a very large number of iterations and more calculation time. For that, we have proposed other procedures to improve the numerical behavior of our algorithm namely: the dynamic choice and the practical choice which replace the theoretical choice. The results obtained are in favor of these two choices. The number of iterations and the calculation time are reduced considerably.*

7. conclusion

In this paper, a new parametric kernel function with a double barrier term (combination of the classic kernel function and a barrier term) is introduced. Based on this function, a class of large and small-update primal–dual interior-point algorithms for $P_*(\kappa)$-linear complementarity problem 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. Finally, we present some numerical results to show the efficiency and the validity of the new proposed kernel functions for this class of problems. As further research, it would be interesting to extend this work to the more general classes of problems such as the horizontal linear complementarity problem (HLCP), nonlinear complementarity problem (NCP), convex problems and second-order cone optimization (SOCO).

References

[1] K. Amini, M.R. Peyghami, Exploring complexity of large update interior-point methods for $P_*(\kappa)$-linear complementarity problem based on kernel function. Applied Mathematics and Computation. **207** (2009) 501–513.

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

[3] Y.Q. Bai, G. Lesaja,C. Roos, A new class of polynomial interior-point algorithms for $P_*(\kappa)$-linear complementarity problems. Pacific Journal of Optimization. **4**(1) (2008) 19–41.

[4] G.M. Cho, M.K. Kim, Y.H. Lee, Complexity of large-update interior point algorithm for $P_*(\kappa)$-linear complementarity problems. Computers and Mathematics with Applications. **53**(6) (2007) 948–960.

[5] G.M. Cho, A new large-update interior point algorithm for $P_*(\kappa)$-linear complementarity problems. Journal of Computational and Applied Mathematics. **216**(1) (2008) 256–278.

[6] G.M. Cho, M.K. Kim, A new large-update interior point algorithm for $P_*(\kappa)$ -LCPs based on kernel functions. Applied Mathematics and Computation. **182**(2) (2006) 1169–1183.

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

[8] L. Derbal, Z. Kebbiche, Theoretical and numerical result for linear optimization problem based on a new kernel function. Journal of Siberian Federal University. Mathematics & Physics, **12**(2) (2019) 160 -172.

[9] L. Derbal, Z. Kebbiche, An efficient parameterized logarithmic kernel function for semidefinite optimization. Acta Mathemacae Applicatae Sinica, English Series, **36**(3) (2020) 753-770.

[10] M. El Ghami, G.Q. Wang, Interior-point methods for $P_*(\kappa)$-linear complementarity problem based on generalized trigonometric barrier function. International Journal of Applied Mathematics, **30**(1) (2017) 11-33.
[11] S. Fathi-Hafshejani, M. Fatemi, M.R. Peygham, An interior-point method for $P_*(\kappa)$-linear complementarity problem based on a trigonometric kernel function. J. Appl. Math. Comput, **48** (2015) 111–128.
[12] S. Fathi-Hafshejani, H. Mansourib, M.R. Peyghami, An interior-point algorithm for $P_*(\kappa)$-linear complementarity problem based on a new trigonometric kernel function.Journal of Mathematical Modeling , **5**(2) (2017) 171-197.
[13] A. Keraghel, Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar, (Ph.D. thesis), Joseph Fourier University, Grenoble I, France, 1989.
[14] M. Kojima, N. Megiddo, T. Noma, A. Yoshise, A unifided approach to interior point algorithms for linear complementarity problems, Lecture Notes in Computer Science, vol. **538**, Springer-Verlag, Berlin, Germany, 1991.
[15] Y.H. Lee, Y.Y. Cho, G.M. Cho, Interior-point algorithms for $P_*(\kappa)$-LCP based on a new class of kernel functions. Journal of Global Optimization, **58**(1) (2014) 137-149.
[16] G. Lesaja, C. Roos, Unified analysis of kernel-based interior-point methods for $P_*(\kappa)$-linear complementarity problems. SIAM Journal on Optimization, **20**(6) (2011) 3014–3039.
[17] G. Lesaja, G.Q. Wang, D.T. Zhu, Interior-point methods for Cartesian $P_*(\kappa)$-linear complementarity problems over symmetric cones based on the eligible kernel functions. Optimization Methods and Software, **27**(4-5) (2012) 827–843.
[18] J. Peng, C. Roos, T. Terlaky, Self-regular functions and new search directions for linear and semidefinite optimization. Mathematical Programming, **93** (1) (2002) 129-171.
[19] M.R. Peyghami, K. Amini, A kernel function based interior-point methods for solving $P_*(\kappa)$-linear complementarity problem. Acta Mathemacae Applicatae Sinica, English Series, **26**(9) (2010) 1761–1778.
[20] Z.G. Qian, Y.Q. Bay, Primal-dual interior-point algorithms with dynamic step size based on kernel functions for linear programming. J. of Shanghai Univ, **9**(5) (2005) 391–396.
[21] C. Roos, T. Terlaky, J.Ph. Vial, Theory and algorithms for linear optimization, in: An interior point Approach. JohnWiley & Sons, Chichester, UK (1997).
[22] G.Q. Wang, Y.Q. Bai, Polynomial interior-point algorithms for $P_*(\kappa)$-horizontal linear complementarity problem. Journal of computational and Applied Mathematics, **233**(2) (2009) 248–263.
[23] G.Q. Wang,Y.Q. Bai, A class of polynomial interior-point algorithms for the cartesian P-Matrix linear complementarity problem over symmetric cones. Journal of Optimization Theory and Application, **152**(3) (2012) 739–772.
[24] G.Q. Wang, C.J. Yu, K.L. Teo, A full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-linear complementarity problem. J. Glob. Optim. **59**(1) (2014) 81–99.
[25] G.Q. Wang, X.J. Fan, D.T. Zhu, D.Z. Wang, New complexity analysis of a full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-LCP. Optim. Lett.**9** (2015) 1105–1119.
[26] S.J. Wright, Primal-Dual Interior-Point Methods. SIAM, Philadelphia, USA, 1997
[27] D. Zhu, M. Zhang, A full-Newton step infeasible interior-point algorithm for $P_*(\kappa)$-linear complementarity problem. J. Syst. Sci. Complex. **27** (2014) 1027–1044.