Filomat 38:26 (2024), 9111–9125
https://doi.org/10.2298/FIL2426111B

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

New kernel function with parametrized default step size for linear complementarity problem

Ayache Benhadid[a,*], Fateh Merahi[b]

[a] *Department of Mathematics, Faculty of Mathematics and Computer Science, Batna 2 University, Batna, Algeria*
[b] *Department of Statistics and Data Science, Faculty of Mathematics and Computer Science, Batna 2 University, Batna, Algeria*

Abstract. Defining new search directions for the primal-dual interior-point algorithm for solving linear complementarity problems mainly depends on kernel functions. In this paper, we introduce a new logarithmic kernel function with a double barrier term and two parameters. The first parameter is used to determine the best complexity for the large-update primal-dual IPMs, while the second parameter is used to control the default step size. These results represent the best-known complexity bound for large-update with a logarithmic boundary term. Finally, we report some numerical results to demonstrate the practical performance of the proposed algorithm with different parameters.

1. Introduction

Linear complementarity problems (LCP) are a general class of mathematical problems that have several important applications in mathematical programming and equilibrium problems. They include linear optimization (LO) problems, convex quadratic optimization (CQO) problems, Nash equilibrium point problems, and so on.

Since the path-breaking paper of Karmarkar [10], kernel functions play an important role in the complexity analysis of the interior point methods (IPMs) for linear complementarity problems (LCP).

In 2001, Peng et al. [15] designed a new paradigm of primal-dual algorithms based on the so-called self-regular proximity functions for LO. The iteration bound was improved by them, and they were able to achieve the best-known complexity results for both large and small update methods. Subsequently, in 2004, Bai et al. [2] proposed a new kernel function with an exponential barrier term and introduced the first new kernel function with a trigonometric barrier term. These functions enjoy useful properties and determine new search directions for primal-dual interior point algorithms. Based on these functions, they obtained the best known complexity results for large-update methods, namely, $\mathbf{O}\left(\sqrt{n}\log n \log \frac{n}{\varepsilon}\right)$ and good numerical results.

In 2008, El Ghami et al. [9] proposed a parameterized kernel function with a logarithmic barrier term. This function generalized the kernel functions given in [7, 16].

2020 *Mathematics Subject Classification*. Primary 90C05; Secondary 90C51, 90C31.
Keywords. Linear complementarity, Kernel function, Interior point methods, Complexity bound.
Received: 06 March 2023; Revised: 15 September 2024; Accepted: 26 September 2024
Communicated by Predrag Stanimirović

* Corresponding author: Ayache Benhadid
Email addresses: a.benhadid@univ-batna2.dz (Ayache Benhadid), f.merahi@univ-batna2.dz (Fateh Merahi)

In 2018, Bouafia et al. [6] proposed a parameterized logarithmic kernel function for primal-dual IPMs whose barrier term is the mean of both barrier terms of Peng et al. [15] and Elghami et al. [9]. They obtained the best-known results in terms of complexity for large and small-update methods. The aim of this paper is to introduce a new kernel function that reduces the number of iterations by involving another parameter β, which allows us to control the default step size α. For this purpose, we generalize the approach of Bouafia et al. [6] by taking the barrier term of our kernel function as a convex combination between the barrier terms of Peng et al. [15] and Elghami et al. [9]. (See Remarkk 3.1, section 3).

The paper is organized as follows. In Section 2, we recall the preliminaries. In Section 3 and 4, we define a new kernel function and give its properties which are essential for the complexity analysis. The estimate of the step size and the decrease behavior of the new barrier function are discussed in Section 5. In section 6 we derive the complexity result for both large-update and small-update methods. In Section 7, we offer some numerical results to show the practical performance of the proposed algorithm with different parameters. The last section ends with concluding remarks.

Some of the notations used throughout the paper are as follows: $\mathbb{R}^n$, $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ denote the set of vectors with n components, the set of nonnegative vectors, and the set of positive vectors, respectively. xs denotes the componentwise product of the vector x and s. $X = diag(x)$ denotes the $n \times n$ diagonal matrix with components of the vector $x \in \mathbb{R}^n$ as the diagonal entries. e denotes the n-dimensional vector of ones.

For $f(x), g(x) : \mathbb{R}^n_{++} \to \mathbb{R}^n_{++}$, $f(x) = \mathbf{O}(g(x))$ if $f(x) \leq C_1 g(x)$ for some positive constant C_1, and $f(x) = \Theta(g(x))$ if $C_2 g(x) \leq f(x) \leq C_3 g(x)$ for some positive constants C_2 and C_3.

2. Preliminaries

In this paper, we consider the Linear Complementarity Problem (LCP) in the standard form: find vectors x and s in real space $\mathbb{R}^n$ that satisfy the following conditions:

$$\begin{cases} s = Mx + b, x \geq 0, \\ xs = 0, s \geq 0, \end{cases} \tag{1}$$

where b is a given vector in $\mathbb{R}^n$ and M is a given $\mathbb{R}^{n\times n}$ real matrix. In this paper, we deal with the special case when the LCP is monotone, i.e., the matrix M is positive semidefinite.

To solve LCP one needs to find a solution of the above system of equations (1) and where xs in the last equation represents the componentwise (Hadamard) product of the vectors x and s.

The general idea is to solve (1) using Newton's method. However, Newton's method can ``get stuck" at the complementarity equation $xs = 0$. Therefore, the main idea of primal-dual interior-point methods is to replace the last equation in (1), the so-called complementarity equation, with the parameterized equation $xs = \mu e$, with parameter $\mu > 0$. So we consider the following system

$$\begin{cases} s = Mx + b, x \geq 0, \\ xs = \mu e, s \geq 0, \end{cases} \tag{2}$$

e is defined as a vector that contains ones of size n. By the last equation, any solution (x, s) of (2) will satisfy $x > 0$ and $s > 0$. Suppose, there exists a point $(x^0, s^0) > 0$ such that

$$Mx^0 + b - s^0 = 0, \tag{3}$$

which means that the interior of the feasible region of (1) is not empty. This assumption is called the interior-point condition (IPC) of the LCP. If the IPC is not satisfied, the modified LCP can be constructed so that it satisfies the IPC. From the solution of the modified LCP, the solution of the original LCP can easily be found. See chapter five in Kojima et al. [11]. Thus, we can, and in this paper, we will always assume that the IPC is satisfied.

It can be shown that for certain classes of matrices, if M has full rank, i.e., $rank(M) = n$, and IPC holds, then the parameterized system (2) has a unique solution for each $\mu > 0$ (see Lemma 4.3 in [11]). This is particularly true for positive semi-definite matrices that we are considering in this work. This solution is denoted as $(x(\mu), s(\mu))$, and we call $(x(\mu), s(\mu))$ the μ-center of (1). The set of μ-centers (with μ running through all positive real numbers) gives a homotopy path, which is called the central path of (1). The importance of the central path for the LP was discovered first by Sonnevend [17] and Megiddo [12] and later generalized to LCP by Kojima et al. [11]. The main property of the central path is that if $\mu \to 0$, then the limit of the central path exists and since the limit points satisfy the complementarity condition, the limit yields the optimal solutions for (1).

This limiting property of the central path leads to the main idea of the iterative methods for solving (1): Trace the central path while reducing μ at each iteration. Theoretically, an exact trace is wanted, but practically it is too inefficient. However, it has been shown that it is only necessary to trace the central path approximately in order to maintain favorable convergence properties of the given algorithms.

We assume that a point (x, s) is "close" to the μ-center, $(x(\mu), s(\mu))$ for some parameter $\mu > 0$. Then, μ is decreased to $\mu_+ := (1 - \theta)\mu$, for some $\theta \in (0, 1)$. Next, we redefine $\mu = \mu_+$, then we solve the following Newton system

$$\begin{cases} -M\Delta x + \Delta s = 0, \\ s\Delta x + x\Delta s = \mu e - xs. \end{cases} \tag{4}$$

Since M has full row rank, the system (4) has a unique solution for any $(x, s) > 0$. The solution $(\Delta x, \Delta s)$ is known as the Newton direction. By taking a step along this search direction, we construct a new ordered pair (x_+, s_+) with

$$x_+ = x + \alpha\Delta x, s_+ = s + \alpha\Delta s, \tag{5}$$

where the step size α satisfies $0 < \alpha \leq 1$, which must be chosen carefully. If needed, we repeat the procedure until we find iterates that are in a certain neighborhood of the μ-center $(x(\mu), s(\mu))$. Then, again, μ is reduced by the factor $1 - \theta$ and Newton's method is applied again targeting the new μ-center, and so on. We repeat this process until μ is small enough, i.e. $n\mu \leq \epsilon$, where ϵ is a small positive number. At this stage in the algorithm, we have found ϵ-approximate solutions of (1).

The scaled vector v and the scaled search directions d_x and d_s are now introduced as follows:

$$v = \sqrt{\frac{xs}{\mu}}, d_x = \frac{v\Delta x}{x}, d_s = \frac{v\Delta s}{s}, \tag{6}$$

where each of the operations is a component-wise product and division. Note that the pair (x, s) coincides with the μ-center $(x(\mu), s(\mu))$ if and only if $v = e$. Note that, if v and the search directions dx, ds are defined as in (6), then the Newton system from (4) can be transformed into the following system:

$$\begin{cases} -\widetilde{M}d_x + d_s = 0, \\ d_x + d_s = v^{-1} - v, \end{cases} \tag{7}$$

where $\widetilde{M} := DMD$, $D := X^{\frac{1}{2}}S^{-\frac{1}{2}}$, $S := \text{diag}(s)$, and $X := \text{diag}(x)$.

One may easily verify that if matrix M is positive semi-definite, then $\widetilde{M}$ is also positive semi-definite.

$$d_x = d_s = 0 \Leftrightarrow v^{-1} - v = 0 \Leftrightarrow v = e. \tag{8}$$

Therefore, we see that $d_x = d_s = 0$ if and only if the pair (x, s) coincides with the μ-center $(x(\mu), s(\mu))$. Unfortunately, d_x and d_s are not, in general, orthogonal vectors, as in the LP case, which will complicate the analysis of the algorithm.

A very important observation is that the right hand side $v^{-1} - v$ in the last equation of (7) equals the negative gradient of the function

$$\Phi(v) = \Phi(x, s; \mu) = \sum_{i=1}^{n} \psi(v_i), \tag{9}$$

where

$$\psi(v_i) = \left(\frac{v_i^2 - 1}{2} - \log v_i \right), \tag{10}$$

which can be written as

$$\begin{cases} -\widetilde{M} d_x + d_s = 0, \\ d_x + d_s = -\nabla\Phi(v). \end{cases} \tag{11}$$

This equation is known as the scaled centering equation. The scaled centering equation basically defines the search directions. An easy verification is that $\nabla^2\Phi(v) = diag(e + v^{-2})$. Since this matrix is positive definite, $\Phi(v)$ is strictly convex. We can see that $\nabla\Phi(v) = 0$, hence $\Phi(v)$ attains its minimal value at $v = e$, with $\Phi(v) = 0$. So, it follows that $\Phi(v)$ is non-negative everywhere and vanishes at $v = e$, which means it vanishes at the μ-center $(x(\mu), s(\mu))$. Therefore, we can conclude that the μ-center $(x(\mu), s(\mu))$ can be characterized as the minimizer of the function $\Phi(v)$. Thus, $\Phi(v)$ serves as a measure of how close (x, s) is to the μ-center.

We use $\Phi(v)$ as the proximity function to measure the distance between the current iterate and the μ-center for a given $\tau > 0$. We also define the norm-based proximity measure, $\delta(v) : \mathbb{R}^n_{++} \to \mathbb{R}_+$, as follows

$$\delta(v) = \frac{1}{2}\|\nabla\Phi(v)\| = \frac{1}{2}\|d_x + d_s\|. \tag{12}$$

We call $\psi(t)$ the kernel function of the logarithmic barrier function $\Phi(v)$. In this paper, we replace $\psi(t)$ by a new kernel function $\psi_\beta(t)$ and $\Phi(v)$ by a new barrier function $\Phi_\beta(v)$, which will be defined in Sect. 3.

The generic primal-dual algorithm can now be formally described. As mentioned, this algorithm follows the central path approximately. Suppose we start with (x, s) close to the μ-center, then μ is reduced to $\mu_+ = (1 - \theta)\mu$. Therefore, the new v becomes $v_+ = \frac{v}{\sqrt{(1-\theta)}}$. As a consequence, $\Phi_\beta(v)$ changes to $\Phi_\beta(v_+)$. The inequality $\Phi_\beta(v) \leq \tau$ means that (x, s) is in a τ-neighborhood of the μ-center $(x(\mu), s(\mu))$, where $\tau > 0$ represents a certain threshold value. Recall that we measure the closeness of (x, s) to the μ-center $(x(\mu), s(\mu))$ by the value of $\Phi_\beta(v)$. However, after the θ-update, the updated $\Phi_\beta(v_+)$ may be greater than τ. If this is the case, we need to perform further steps to reduce $\Phi_\beta(v_+)$ and get closer to the new μ-center. In other words, we need to get back to the τ-neighborhood of the new μ-center.

After finding d_x and d_s from (11), we can find the original directions Δx and Δs from (4). Alternatively, Δx and Δs can be found directly from the following system

$$\begin{cases} -M\Delta x + \Delta s = 0, \\ s\Delta x + x\Delta s = -\mu v \nabla\Phi(v). \end{cases} \tag{13}$$

To accomplish this, we need to first find the direction Δx and Δs by solving the Newton system (13). We update x and s using a chosen step size α and the recently found search directions Δx and Δs, respectively. This process is repeated until $\Phi_\beta(v) \leq \tau$, upon which the process begins again. We begin again by reducing μ and updating v, and so on until we have a μ-center that is ϵ-close to the actual solution. The generic form of the algorithm is shown in Fig.1. In the sequel, we will refer to it as simply the Generic Algorithm.

```
Generic Primal-Dual Algorithm for LCP
Input:
 Determine input parameters:
 a threshold parameter $\tau > 0$,
 an accuracy parameter $\epsilon > 0$,
 a fixed barrier update parameter $\theta, 0 < \theta < 1$,
begin
Set $(x_0, s_0, \mu_0) > 0$ so that the IPC is satisfied, while $n\mu \geq \epsilon$ do
$\mu = (1-\theta)\mu$,
$v = \sqrt{\frac{xs}{\mu}}$ begin (outer iteration)
 while $\Phi_\beta(v)(x, s; \mu) > \tau$ do
 begin (inner iteration)
 solve the system (13) , $\Phi(v)$ replaced by $\Phi_\beta(v)$ to obtain $(\Delta x, \Delta s)$,
 choose a suitable step size $\alpha$,
 Update $x = x + \alpha\Delta x, s = s + \alpha\Delta s$
 Update $v = \sqrt{\frac{xs}{\mu}}$
 end (inner iteration)
 end (outer iteration)
end .
```

Fig.1 Generic Primal-Dual Algorithm for LCP.

We want to optimize the algorithm by minimizing the number of iterates in the algorithm. To do this we must carefully choose the parameters τ, θ, and the step size α. Choosing the barrier update parameter θ is very important in application and theory. If θ is a constant number which is independent of the dimension n of the problem, i.e., $\theta = \mathbf{\Theta}(1)$, then the algorithm is called a large update method. If θ depends on the dimension n of the problem, then we call the algorithm a small update method. In this case, θ is usually chosen to be the following: $\theta = \mathbf{\Theta}\left(\frac{1}{\sqrt{n}}\right)$.

Choosing the step size, $\alpha > 0$, is another key step in obtaining good convergence properties of the algorithm. It must be set in such a way that the closeness of the iterates to the current μ-center improves by a sufficient amount.

In this paper, we define a new logarithmic kernel function with two parameters for LCP and prove that for all β, the corresponding algorithm has $\mathbf{O}\left(qn^{\frac{q+1}{2q}} \log\left(\frac{n}{\epsilon}\right)\right)$ complexity bound for the large-update method and $\mathbf{O}\left(q^2 \sqrt{n} \log\left(\frac{n}{\epsilon}\right)\right)$ for the small-update method. Another interesting choice is q and β, which minimizes the iteration complexity bound. In fact, if we take $q = \frac{\log n}{2}$ and β in the neighborhood of 0, we obtain good convergence properties of the algorithm with the best known complexity bound for large-update methods, namely $\mathbf{O}\left(\sqrt{n} \log n \log \frac{n}{\varepsilon}\right)$.

3. The New Kernel Function and Its Properties

In this section, a new kernel function and its properties are provided. Let us define the new univariate function.

$$\psi_\beta(t) = \frac{1}{2}\left(t^2 - 1\right) - \beta \log(t) + (1-\beta)\frac{t^{1-q} - 1}{q-1}, q > 1, \beta \in [0, 1]. \tag{14}$$

The parameters β and q are two constants and play an important role in the analysis of proposed methods. It follows that $\psi_\beta(1) = \psi'_\beta(1) = 0$ and $t \to 0^+$ or $t \to +\infty$, then $\psi(t) \to +\infty$. As we need the first three derivatives of $\psi_\beta(t)$ with respect to t frequently, we provide them as follows:

$$\psi_\beta^{'}(t) = t - \beta t^{-1} - (1-\beta)t^{-q}, \tag{15}$$
$$\psi_\beta^{''}(t) = 1 + \beta t^{-2} + (1-\beta)qt^{-(q+1)}, \tag{16}$$
$$\psi_\beta^{'''}(t) = -2\beta t^{-3} - (1-\beta)q(q+1)t^{-(q+2)}. \tag{17}$$

We will now discuss some special cases.

Speciale cases:

a) If $\beta = 0$, then the kernel function ψ_β is equivalent to Peng's kernel $\psi_p(t) = \frac{1}{2}\left(t^2-1\right) + \frac{t^{1-q}-1}{q-1}$ introduced and studied by Peng et al. [15].
b) If $\beta = 1$, then the kernel function ψ_β reduces to the classical kernel function $\psi_{EL}(t) = \frac{1}{2}\left(t^2-1\right) - \log(t)$ introduced and studied by El Ghami et al. [9].
c) If $\beta = \frac{1}{2}$, then the kernel function ψ_β is equivalent to Bouafia's kernel ψ_{LB} introduced and studied by Bouaafia et al. [6].
d) If $\beta \in [0,1] \setminus \left\{0, \frac{1}{2}, 1\right\}$, then the kernel function ψ_β appears to be a new one.

Remark 3.1. *We can see that the barrier term Bar_β of our kernel function ψ_β is the convex combination between the barrier term $Bar_{EL}(t) = -\log(t)$ of $\psi_{EL}(t)$ El Ghami et al. [9] and the barrier term $Bar_p(t) = \frac{t^{1-q}-1}{q-1}$ of $\psi_p(t)$ Peng et al. [15], Indeed*

$$\begin{aligned} Bar_\beta(t) &= \beta Bar_{EL}(t) + (1-\beta)Bar_p(t) \\ &= -\beta\log(t) + (1-\beta)\frac{t^{1-q}-1}{q-1}. \end{aligned}$$

4. Eligibility of the New Kernel Function

The following lemma plays an important role in the analysis of the kernel function and serves to prove that the new kernel function (14) is efficient.

Lemma 4.1. *Let $\psi_\beta(t)$ be as defined in* (14) *and $t > 0$. Then,*

$$\psi_\beta^{''}(t) > 1, \tag{18}$$
$$\psi_\beta^{'''}(t) < 0, \tag{19}$$
$$t\psi_\beta^{''}(t) - \psi_\beta^{'}(t) > 0, \tag{20}$$
$$t\psi_\beta^{''}(t) + \psi_\beta^{'}(t) > 0. \tag{21}$$

Proof. From (16) and (17), it is obvious that (18) and (19) are satisfied.
From (15) and (16), we have the following:

$$t\psi_\beta^{''}(t) - \psi_\beta^{'}(t) = 2t + (1-\beta)(q-1)t^{-q} > 0,$$

and

$$t\psi_\beta^{''}(t) + \psi_\beta^{'}(t) = 2\beta t^{-1} + (1-\beta)(q+1)t^{-q} > 0,$$

which proves (20) and (21). □

The last property (21) is equivalent to

$$\psi_\beta(\sqrt{t_1 t_2}) \leq \frac{1}{2}\left(\psi_\beta(t_2) + \psi_\beta(t_2)\right), \text{ for any } t_1, t_2 \geq 0, \tag{22}$$

and it was demonstrated by several researchers (see Bai et al. [2] and Megiddo [12]). Thus $\psi_\beta(t)$ is an eligible kernel function.

Lemma 4.2. *For $\psi_\beta(t)$, we have*

$$\frac{1}{2}(t-1)^2 \leq \psi_\beta(t) \leq \frac{1}{2}\left[\psi_\beta^{'}(t)\right]^2, t > 0. \tag{23}$$

$$\psi_\beta(t) \leq [1 + \frac{1}{2}(1-\beta)(q-1)](t-1)^2, t > 1. \tag{24}$$

Proof. For (23), using (18), we have

$$\psi_\beta(t) = \int_1^t \int_1^x \psi_\beta^{''}(y)dydx \geq \int_1^t \int_1^x 1dydx = \frac{1}{2}(t-1)^2.$$

$$\begin{aligned}\psi_\beta(t) &= \int_1^t \int_1^x \psi_\beta^{''}(y)dydx \leq \int_1^t \int_1^x \psi_\beta^{''}(y)\psi_\beta^{''}(x)dydx \\ &= \int_1^t \psi_\beta^{''}(x)\psi_\beta^{'}(x)dx \\ &= \int_1^t \psi_\beta^{'}(x)d\psi_\beta^{'}(x) = \frac{1}{2}\left[\psi_\beta^{'}(t)\right]^2.\end{aligned}$$

For (24), by using Taylor's formula, we have

$$\begin{aligned}\psi_\beta(t) &= \psi_\beta(1) + \psi_\beta^{'}(1)(t-1) + \frac{1}{2}\psi_\beta^{''}(1)(t-1)^2 + \frac{1}{6}\psi_\beta^{'''}(\xi)(t-1)^3 \\ &= \frac{1}{2}\psi_\beta^{''}(1)(t-1)^2 + \frac{1}{6}\psi_\beta^{'''}(\xi)(t-1)^3 \\ &\leq \frac{1}{2}\psi_\beta^{''}(1)(t-1)^2 \\ &= [1 + \frac{1}{2}(1-\beta)(q-1)](t-1)^2,\end{aligned}$$

for some ξ, $1 \leq \xi \leq t$. □

Lemma 4.3. *For $\psi_\beta(t)$, we have*

$$1 + \sqrt{\frac{2s}{2 + (1-\beta)(q-1)}} \leq \sigma(s) \leq 1 + \sqrt{2s}, s \geq 0. \tag{25}$$

$$\rho(s) \geq \left[\frac{1-\beta}{2s+1}\right]^{\frac{1}{q}}, s > 0, \tag{26}$$

where $\sigma : [0, \infty[\to [1, +\infty[$ is the inverse function of $\psi_\beta(t)$ for $t \geq 1$, and $\rho : [0, \infty[\to]0, 1]$ is the inverse function of $-\frac{1}{2}\psi_\beta^{'}(t)$ for all $t \in]0, 1]$.

Proof. For (25), let $s = \psi_\beta(t)$, $t \geq 1$. By (23), we have $\psi_\beta(t) \geq \frac{1}{2}(t-1)^2$, $t \geq 1$. This implies that $t = \sigma(s) \leq 1 + \sqrt{2s}$. By (24), we have

$$s = \psi_\beta(t) \leq [1 + \frac{1}{2}(1-\beta)(q-1)](t-1)^2, t \geq 1, \text{so } t = \sigma(s) \geq 1 + \sqrt{\frac{2s}{2+(1-\beta)(q-1)}}.$$

For (26), let $s = -\frac{1}{2}\psi'_\beta(t)$, $t \in]0, 1]$. By the definition of $\psi'_\beta(t)$, we have

$$2s = -t + \beta t^{-1} + (1-\beta)t^{-q} \geq -1 + (1-\beta)t^{-q},$$

which implies that $t = \rho(s) \geq \left[\frac{1-\beta}{2s+1}\right]^{\frac{1}{q}}$. □

Lemma 4.4. *Let $\sigma : [0, \infty[\to [1, +\infty[$ be the inverse function of $\psi_\beta(t)$ for $t \geq 1$. Then we have*

$$\Phi_\beta(\beta v) \leq n\psi_\beta\left(\beta\sigma\left(\frac{\Phi_\beta(v)}{n}\right)\right), v \in \mathbb{R}^*, \beta \geq 1.$$

Proof. Using (19) and (20), and Lemma 2.4 in [2], we can obtain the result. □

Lemma 4.5. *Let $0 \leq \theta < 1$, $v_+ = \frac{v}{\sqrt{1-\theta}}$. If $\Phi_\beta(v) \leq \tau$, then we have the following upper bounds on the value of $\Phi_\beta(v_+)$ after a μ-update: $\Phi_\beta(v_+) \leq L_i$, $i = 1, 2$, where*

$$L_1 := \frac{\theta n + 2\tau + 2\sqrt{2\tau n}}{2(1-\theta)},$$
$$L_2 := \frac{q+1}{2(1-\theta)}\left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2.$$

Proof. For L_1, since $\frac{1}{\sqrt{1-\theta}} \geq 1$ and $\sigma\left(\frac{\Phi_\beta(v)}{n}\right) \geq 1$, then $\frac{\sigma\left(\frac{\Phi_\beta(v)}{n}\right)}{\sqrt{1-\theta}} \geq 1$. And for $t \geq 1$, we have $\psi_\beta(t) \leq \frac{(t^2-1)}{2}$. Using Lemma 4.4 with $\beta = \frac{1}{\sqrt{1-\theta}}$, (25), and $\Phi_\beta(v) \leq \tau$, we have

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

For L_2, by (24), with Lemma 4.4, we get the result. □

5. Analysis of Algorithm

5.1. Determining a default step size

In the following section, we will compute a proper default step size α and the decrease of the barrier function during an inner iteration, and provide the complexity results of the algorithm.

Lemma 5.1. *Let $\delta(v)$ be as defined in* (12)*. Then we have*

$$\delta(v) \geq \sqrt{\frac{\Phi_\beta(v)}{2}}. \tag{27}$$

Proof. Using (23), we have

$$\Phi_\beta(v) = \sum_{i=1}^{n} \psi_\beta(v_i) \leq \sum_{i=1}^{n} \frac{1}{2}\left[\psi'_\beta(v_i)\right]^2 = \frac{1}{2}\|\nabla\Phi_\beta(v)\|^2 = 2\delta(v)^2,$$

so $\delta(v) \geq \sqrt{\frac{1}{2}\Phi_\beta(v)}$.
□

Throughout the paper, we assume that $\Phi_\beta(v) \geq \tau \geq 1$, and we have $\delta(v) \geq \sqrt{\frac{1}{2}}$.

Lemma 5.2. *[Bai et al.[2]] The largest step size $\overline{\alpha}$ is given by*

$$\overline{\alpha} = \frac{\rho(\delta) - \rho(2\delta)}{2\delta} \geq \frac{1}{\psi''_\beta(\rho(2\delta))}.$$

Lemma 5.3. *The largest step size $\overline{\alpha}$ verifies*

$$\overline{\alpha} \geq \frac{1}{1 + [1 + (1-\beta)(q-1)](\frac{4\delta+1}{1-\beta})^{\frac{q+1}{q}}}.$$

Proof. Using Lemma 5.2, (16) and (26), we have

$$\begin{aligned}
\overline{\alpha} &\geq \frac{1}{\psi''_\beta(\rho(2\delta))} \\
&= \frac{1}{1 + \beta[\rho(2\delta)]^{-2} + (1-\beta)q[\rho(2\delta)]^{-(q+1)}} \\
&\geq \frac{1}{1 + \beta[\frac{4\delta+1}{1-\beta}]^{\frac{2}{q}} + (1-\beta)q[\frac{4\delta+1}{1-\beta}]^{\frac{q+1}{q}}} \\
&\geq \frac{1}{1 + \beta[\frac{4\delta+1}{1-\beta}]^{\frac{2}{q}} + (1-\beta)q[\frac{4\delta+1}{1-\beta}]^{\frac{q+1}{q}}} \\
&\geq \frac{1}{1 + \beta[\frac{4\delta+1}{1-\beta}]^{\frac{q+1}{q}} + (1-\beta)q[\frac{4\delta+1}{1-\beta}]^{\frac{q+1}{q}}} \\
&\geq \frac{1}{1 + [1 + (1-\beta)(q-1)](\frac{4\delta+1}{1-\beta})^{\frac{q+1}{q}}}.
\end{aligned}$$

□

Denoting

$$\widetilde{\alpha} = \frac{1}{1 + [1 + (1-\beta)(q-1)](\frac{4\delta+1}{1-\beta})^{\frac{q+1}{q}}}, \tag{28}$$

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

Remark 5.4 (Analysis of the default step size). *Note that if we increase the value of β ($\beta \overset{<}{\to} 1$), the denominator value is also increased, which affects $\widetilde{\alpha}$ and makes it close to zero, so the number of iterations increases. We conclude that for small values of β (β in the neighborhood of 0), we get the best approximation of $\widetilde{\alpha}$.*

5.2. Decrease in the proximity function during an inner iteration

Lemma 5.5. *[Lemma 4.5 in [2]] If the step size α satisfies $\alpha \leq \overline{\alpha}$, then*

$$f(\alpha) \leq -\alpha\delta^2,$$

where $f(\alpha) = \Phi(v_+) - \Phi(v)$, i.e $f(\alpha)$ is the difference of proximities between a new iterate and a current iterate for fixed μ.

Theorem 5.6. *Let $\widetilde{\alpha}$ be the default step size as defined in* (28)*. Then, we have*

$$f(\widetilde{\alpha}) \leq -\frac{(1-\beta)^2\Phi_\beta(v)^{\frac{q-1}{2q}}}{36\sqrt{2}(q+1)}. \tag{29}$$

Proof. Since $\Phi_\beta(v) \geq 1$, then from (27), we have $2\delta > 1$. Using Lemma 5.5 (Lemma 4.5 in [2]) with $\alpha = \widetilde{\alpha}$ and (28), we have

$$\begin{aligned}
f(\widetilde{\alpha}) \leq -\widetilde{\alpha}\delta^2 &= -\frac{\delta^2}{1+[1+(1-\beta)(q-1)](\frac{4\delta+1}{1-\beta})^{\frac{q+1}{q}}} \\
&\leq -\frac{\delta^2}{(2\delta)^{\frac{q+1}{q}}+[1+(1-\beta)(q-1)](\frac{6\delta}{1-\beta})^{\frac{q+1}{q}}} \\
&\leq -\frac{\delta^2}{(2)^2(\delta)^{\frac{q+1}{q}}+[1+(1-\beta)(q-1)](\frac{6}{1-\beta})^2(\delta)^{\frac{q+1}{q}}} \\
&\leq -\frac{(1-\beta)^2\delta^{\frac{q-1}{q}}}{4(1-\beta)^2+[1+(1-\beta)(q-1)]36} \\
&\leq -\frac{(1-\beta)^2\delta^{\frac{q-1}{q}}}{4+[1+(1-\beta)(q-1)]36} \\
&\leq -\frac{(1-\beta)^2\Phi_\beta(v)^{\frac{q-1}{2q}}}{36\sqrt{2}(q+1)}.
\end{aligned}$$

□

6. Iteration Complexity

6.1. Inner iteration bound

After the update of μ to $(1-\theta)\mu$, we have

$$\Phi_\beta(v_+) \leq L_i,\ i = 1, 2.$$

We need to count how many inner iterations are required to return to the situation where $\Phi_\beta(v_+) \leq \tau$. We denote the value of $\Phi_\beta(v)$ after the μ update as (L_i), the subsequent values in the same outer iteration are denoted by $\left(\Phi_\beta\right)_k$, $k = 1, 2, ..., K$, where K denotes the total number of inner iterations in the outer iteration. The decrease in each inner iteration is given by (29). In [2], we can find the appropriate values of κ and $\gamma \in]0, 1]$ which are given respectively by

$$\kappa = \frac{(1-\beta)^2}{36\sqrt{2}(q+1)}, \gamma = 1 - \frac{q-1}{2q} = \frac{q+1}{2q}.$$

Lemma 6.1. *In the outer iteration, K is the total number of inner iterations. Then we have*

$$K \leq \frac{72\sqrt{2}q}{(1-\beta)^2}\,[L_i]^{\frac{q+1}{2q}}.$$

Proof. Lemma 1.3.2 in [14] allows us to have

$$K \leq \frac{[L_i]^{\gamma}}{\kappa\gamma} = \frac{72\sqrt{2}q}{(1-\beta)^2}\,[L_i]^{\frac{q+1}{2q}}.$$

□

6.2. Total iteration bound

The number of outer iterations is bounded above by $\frac{\log(\frac{n}{\epsilon})}{\theta}$ (see Roos et al. [16] Lemma II.17, page 116). An upper bound for the total number of iterations is obtained by multiplying the number of outer iterations by the number of inner iterations, specifically,

$$\frac{72\sqrt{2}q}{(1-\beta)^2}\,[L_i]^{\frac{q+1}{2q}}\,\frac{\log\left(\frac{n}{\epsilon}\right)}{\theta}. \tag{30}$$

For large-update methods, $\tau = \mathbf{O}(n)$, $\theta = \Theta(1)$ with upper bounds L_1, we have

$$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right) \text{ iterations complexity.}$$

For small-update methods: $\tau = \mathbf{O}(1)$ and $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$ with upper bounds $L_2 = \mathbf{O}(q)$, we get the following iteration bound

$$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right) \text{ iterations complexity.}$$

In the following section, we present some numerical results to demonstrate the efficiency of the proposed algorithm.

7. Numerical Results

To illustrate the efficiency of the new kernel function proposed in this article, we will compare the performance of our proposed kernel function ψ_β defined in (14) with both kernel functions $\psi_{LB}(t) = \frac{1}{2}\left(t^2-1\right) - \frac{1}{2}\log(t) + \frac{t^{1-q}-1}{2(q-1)}$ and $\psi_{EL}(t) = \frac{1}{2}\left(t^2-1\right) - \log(t)$ which are introduced by Bouafia et al. [6] and El Ghami et al. [9] respectively. In this section we present some numerical results. Therefore, we will discuss the numerical implementation of this algorithm applied to problems related to LCPs in the standard form given by (1). Our numerical example serves to demonstrate the influence of the parameter β included in our proposed kernel function on the number of iterations and to prove and evaluate the effectiveness of our new kernel function ψ_β on this influence behavior of the algorithm. To solve the problem we used Matlab 2013b. We have the tolerance $\epsilon = 10^{-4}$, $\mu_0 = 1.5$, $\theta \in \{0.30, 0.95\}$ and $\tau = n$ where n denotes the number of variables. For the kernel functions, we take the best choice of q, i.e., $q = \frac{\log n}{2}$. For the default step size α we choose $\widetilde{\alpha}_{EL} = \frac{1}{2(1+4\delta)^2}$ according to ψ_{EL} as in [9], $\widetilde{\alpha}_{LB} = \frac{2}{2+(q+1)(8\delta+2)^{\frac{q+1}{q}}}$ according to ψ_{LB} as in [6], and $\widetilde{\alpha}_\beta = \frac{1}{1+[1+(1-\beta)(q-1)](\frac{4\delta+1}{1-\beta})^{\frac{q+1}{q}}}$ which is much the same as the previous kernel functions. In the results tables, we used *Inner It* to represent the number of inner iterations required to obtain the optimal solution, and we used *Outer It* to represent the number of total outer iterations required to obtain the optimal solution.

Example 7.1.

$$\text{Consider } M = \begin{pmatrix} 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, b = (-14, -13, -10, -9, -5, -4, -1, 0).$$

The numerical results are shown in Table 1 below.

Kernel functions	*Large update*	*Small update*	θ	*Inner It.*	*Outer It.*
ψ_{EL}	$\mathbf{O}\left(n\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	7928	34
			0.95	1531	5
$\psi_\beta, (\beta = 0.9)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	334280	34
			0.95	62534	5
$\psi_\beta, (\beta = 0.7)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	39204	34
			0.95	7314	5
ψ_{LB}	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	14614	34
			0.95	2714	5
$\psi_\beta, (\beta = 0.3)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	7591	34
			0.95	1418	5
$\psi_\beta, (\beta = 0.2)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	5905	34
			0.95	1097	5
$\psi_\beta, (\beta = 0.1)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	4730	34
			0.95	876	5
$\psi_\beta, (\beta = 0.05)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	4275	34
			0.95	791	5
$\psi_\beta, (\beta = 0.01)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	3962	34
			0.95	732	5

Table 1. Number of iterations with the kernel functions ψ_{EL}, ψ_{LB}, ψ_β for Example 7.1.

When $\beta = 0.3, 0.2, 0.1, 0.05, 0.01$, Table 1 shows that the number of iterations to execute the algorithm based on the kernel function ψ_β is smaller than the other kernel functions shown with $\theta \in \{0.30, 0.95\}$. We conclude that for small values of β near 0, our proposed kernel functions produce better results than the kernel functions ψ_{LB} and ψ_β. (See Figure 1).

Example 7.2.

$$M=\begin{pmatrix}1&2&2&2&2&2&2&2&2&2&2&2\\0&2&2&2&2&2&2&2&2&2&2&2\\0&0&1&2&2&2&2&2&2&2&2&2\\0&0&0&2&2&2&2&2&2&2&2&2\\0&0&0&0&1&2&2&2&2&2&2&2\\0&0&0&0&0&2&2&2&2&2&2&2\\0&0&0&0&0&0&1&2&2&2&2&2\\0&0&0&0&0&0&0&2&2&2&2&2\\0&0&0&0&0&0&0&0&1&2&2&2\\0&0&0&0&0&0&0&0&0&2&2&2\\0&0&0&0&0&0&0&0&0&0&1&2\\0&0&0&0&0&0&0&0&0&0&0&2\end{pmatrix},$$

b=(-22,-21,-18,-17,-14,-13,-9,-8,-5,-4,-1,0).

Table 2 below lists the numerical results.

Kernel functions	*Large update*	*Small update*	θ	*Inner It.*	*Outer It.*
ψ_{EL}	$\mathbf{O}\left(n\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	13802	35
			0.95	5091	6
$\psi_\beta, (\beta = 0.9)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	280358	35
			0.95	87315	6
$\psi_\beta, (\beta = 0.7)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	41083	35
			0.95	12649	6
ψ_{LB}	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	17677	35
			0.95	*5292*	*6*
$\psi_\beta, (\beta = 0.3)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	1001	35
			0.95	3030	6
$\psi_\beta, (\beta = 0.2)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	8541	35
			0.95	2440	6
$\psi_\beta, (\beta = 0.1)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	6581	35
			0.95	2019	6
$\psi_\beta, (\beta = 0.05)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	6552	35
			0.95	1855	6
$\psi_\beta, (\beta = 0.01)$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	*0.30*	6403	35
			0.95	1737	6

Table 2. Number of iterations with the kernel functions ψ_{EL}, ψ_{LB}, ψ_β for Example 7.2.

As in Example 7.1, the results show that the function ψ_β ($\beta = 0.3, 0.2, 0.1, 0.05, 0.01$) the number of iterations is reduced compared to the other kernel functions, as shown in Table 1 for $\theta \in \{0.30, 0.95\}$. It can be concluded that the new kernel function produces better results than the kernel functions ψ_{EL} and ψ_{LB} when the parameter β taken into account. (See Figure 2).

7.1. Discussion

Based on the obtained results of Tables 1, 2 for Example 7.1 and Example 7.2, the following comments are concluded:
As the value of β increases to 0, the number of iterations increases due to the increased denominator value, which affects $\widetilde{\alpha}$ and brings it close to zero. This increases the number of iterations.
The algorithm based on the kernel function ψ_β provides the best number of iterations when the parameter β decreases to 0. We conclude that for small values of β (β near 0), we get the best approximation for $\widetilde{\alpha}$. This means that the number of iterations is sensitive to the changes in β involved in the proposed kernel function ψ_β.

The number of inner iterations with respect to the parameter β using the kernel function ψ_β for $\theta \in \{0.30, 0.95\}$ is shown in Figures 1, 2 corresponding to Examples 7.1 and 7.2, respectively.

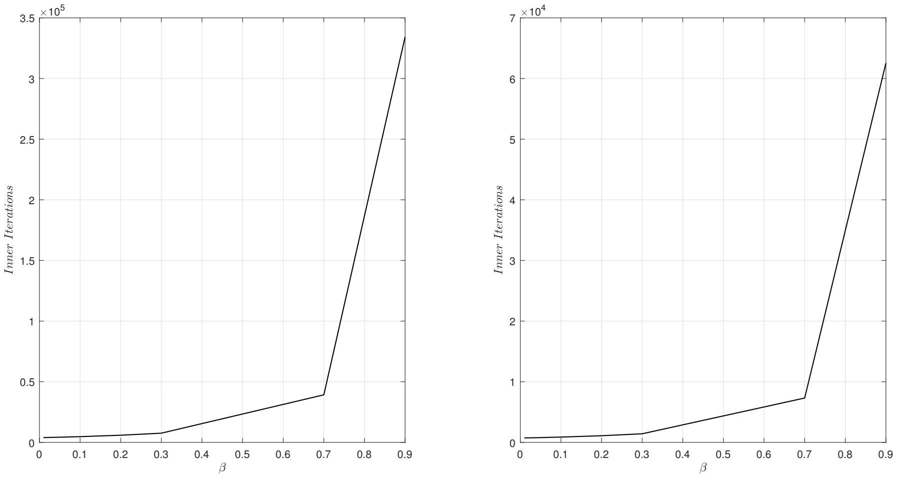

Figure 1: Number of inner iterations with respect to β according to Example 7.1 for $\theta = 0.30$ (left) and $\theta = 0.95$ (right).

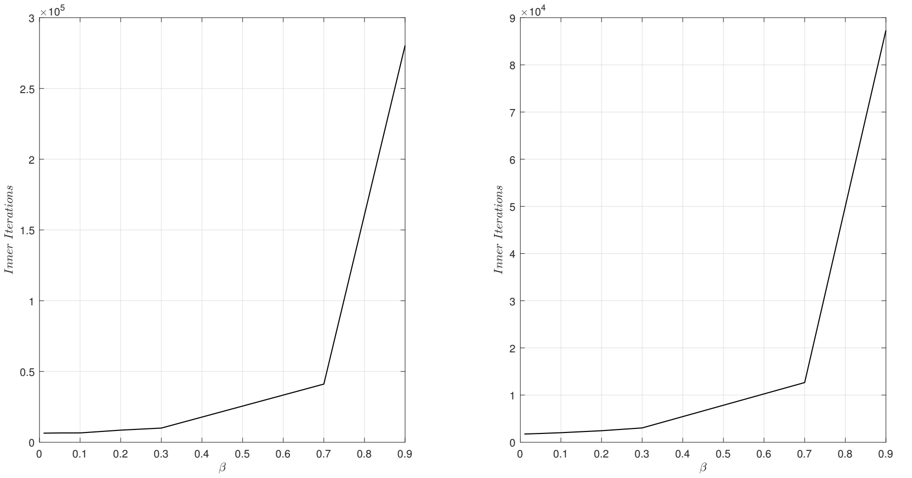

Figure 2: Number of inner iterations with respect to β according to Examples 7.2 for $\theta = 0.30$ (left) and $\theta = 0.95$ (right).

8. Conclusion

In this work, we have proven that for all β in $]0,1[$ the iteration bound of a large-update interior point method based on the kernel function considered in this work is $\mathbf{O}\left(qn^{\frac{q+1}{2q}}\log\left(\frac{n}{\epsilon}\right)\right)$ and $\mathbf{O}\left(q^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$ for the small update method. Another interesting choice is q and β, which minimize the iteration complexity bound. If we assume $q = \frac{\log n}{2}$ and β close to 0, we actually get good convergence properties of the algorithm with the best-known complexity bound for large-update methods, namely $\mathbf{O}\left(\sqrt{n}\log(n)\log\left(\frac{n}{\varepsilon}\right)\right)$. These results represent the best-known complexity bound for large-update with a logarithmic barrier term and are more efficient compared to other logarithmic kernels.

Acknowledgement

The authors are grateful to the handling editor and the anonymous referees for their helpful comments that improved the quality of the paper.

References

[1] Y. Q. Bai, C. Roos, *A primal-dual interior point method based on a new kernel function with linear growth rate*, In: Proceedings of the 9th Australian Optimization Day, Perth, 2000, Australia.

[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 J. Optim. **15** (2004) 101–128.

[3] A. Benhadid, K. Saoudi, *A new parameterized logarithmic kernel function for linear optimization with a double barrier term yielding the best known iteration bound*, Commun. Math. **28** (2020), 27–41.

[4] A. Benhadid, K. Saoudi, F. Merahi, *An interior point approach for linear complementarity problem using new parametrized kernel function*, Optimization. **71** (2022), 4403–4422.

[5] A. Benhadid, F. Merahi, *Complexity analysis of an interior-point algorithm for linear optimization based on a new parametric kernel function with a double barrier term*, Numer. Algebra Control Optim, **13(2)** (2023), 224–238.

[6] M. Bouaafia, D. Benterki, Y. Adnan, *An efficient parameterized logarithmic kernel function for linear optimization*, Optim. Lett, **12** (2018), 1079–1097.

[7] M. El Ghami, *New Primal-Dual Interior-Point Methods Based on Kernel Functions*, PhD Thesis, TU Delft, The Netherlands, 2005.

[8] M. El Ghami, Z. A. Guennoun, S. Bouali, T. Steihaug, *Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term*, J. Comput. Appl. Math. **236** (2012), 3613–3623.

[9] M. El Ghami, I. D. Ivanov, C. Roos, T. Steihaug, *A polynomial-time algorithm for LO based on generalized logarithmic barrier functions*, Int. J. Appl. Math. 21 (2008), 99–115.

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

[11] M. Kojima, N. Megiddo, T. Noma, A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, (1st edition), Springer Berlin, Heidelberg, 1991.

[12] N. Megiddo, *Pathways to the optimal set in linear programming*, In: Megiddo, N. (ed.) Progress in Mathematical Programming: Interior Point and Related Methods, 131–158. Springer, New York (1989).

[13] J. Peng, C. Roos, T. Terlaky, *A new class of polynomial primal–dual methods for linear and semidefinite optimization*, European J. Oper. Res. **143** (2002), 234–256.

[14] J. Peng, C. Roos, T. Terlaky, *Self-Regularity: A New Paradigm for Primal-Dual Interior Point Algorithms*, Princeton University Press, 2002.

[15] 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.

[16] C. Roos, T. Terlaky, J. P. Vial, *Theory and Algorithms for Linear Optimization, An Interior Point Approach*, (1st edition), Wiley, Chichester, 1997.

[17] G. Sonnevend, *An "analytical centre" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming.*, In: Prékopa, A., Szelezsáan, J., Strazicky, B. (eds) System Modelling and Optimization. Lecture Notes in Control and Information Sciences, **84**. Springer, 1986.

[18] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, (1st edition), Wiley, Chichester, 1997.